Métodos en Biomatemática II: Ecuaciones diferenciales

con wxMaxima

Rafael Lahoz-Beltrá

METODOS EN BIOMATEMATICA II: ECUACIONES DIFERENCIALES

R. Lahoz-Beltra

Departamento de Matemática Aplicada (Biomatemática), Facultad de Biología,
C/ José Antonio Novais, 2
Universidad Complutense de Madrid. 28040 Madrid

Introducción

En este manual, y haciendo uso del sistema de álgebra computacional wxMaxima [1], se describen técnicas y métodos fundamentales en Biología Matemática, en particular algunos modelos clásicos de ecuaciones diferenciales ordinarias (EDOs) así como la metodología para resolver con wxMaxima sistemas de dos y tres ecuaciones diferenciales lineales de primer orden. Se asume que el lector conoce la teoría y los razonamientos, explicándose únicamente los procedimientos y rutinas de cálculo con wxMaxima. En cualquier caso, en el libro [2] el lector encontrará información suficiente a nivel de divulgación sobre los aspectos biológicos de algunos modelos matemáticos clásicos en Biología, por ejemplo el modelo de Verhulst, Leslie, el caos en la ecuación logística, el modelo predador presa de Volterra-Lotka, los fractales, álgebra de matrices, etc. entre otros; incluyéndose un bosquejo histórico sobre los orígenes de la Biomatemática. El libro [3] expone en toda su extensión y con suficiente profundidad el concepto de simulación clásica del que se hace uso en este manual. Asimismo, los libros [4, 5] son de interés para un estudio en profundidad de los aspectos formales, es decir matemáticos, de las EDOs y de los sistemas de ecuaciones diferenciales. En resumen, el manual es una colección de protocolos de computación simbólica con los que haciendo uso del entorno wxMaxima es posible resolver toda una serie de casos prácticos en el “Laboratorio de Biomatemática” [6].

I. Métodos para la resolución con wxMaxima de ecuaciones diferenciales ordinarias (EDOs) en Biología

A continuación se describen algunos de los métodos y técnicas con EDOs que son habituales en Biomatemática.

I.1. Estudio general de un fenómeno con EDOs

Supóngase que un biólogo desea estudiar un fenómeno que responde a una cierta EDO:

$$y' = k_1 y - k_2 y$$

A continuación, explicaremos los pasos a seguir para estudiar dicho fenómeno con wxMaxima:

A. Parámetros del modelo y ecuación del modelo.

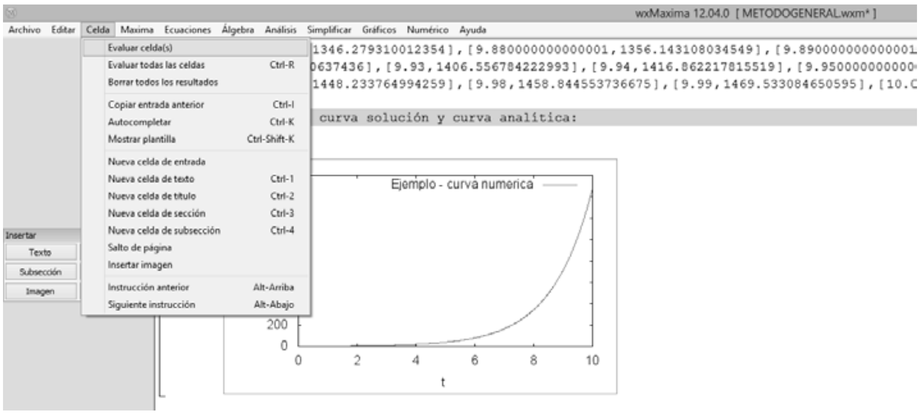
Asumiremos que los **parámetros** k_1 y k_2 son conocidos así como las **condiciones iniciales** y_0. Para escribir con wxMaxima la EDO lo haremos escribiendo **y'** con la orden:

'diff(y,t)

(NOTA: en el teclado español utilice el apóstrofe **'** de la tecla con el símbolo **?**). A continuación, asignaremos un nombre a la EDO, por ejemplo **ecuacion** (no escribir acento). Utilizaremos, según la sintaxis que se muestra más abajo, las órdenes **ode2** para obtener la *solución general* de la EDO y la orden **ic1** para la *solución particular*:

```
Parámetros del modelo:
 -->  k1:0.85; k2:0.12; y0:1;
Ecuación del modelo:
 -->  ecuacion:'diff(y,t)=k1*y-k2*y;
 -->  ode2(ecuacion,y,t);
 -->  ic1(ode2(ecuacion,y,t),t=0,y=y0);
```

Una vez evaluadas las celdas se obtienen los resultados que se muestran más abajo. Evaluaremos a lo largo del ejemplo las celdas, una a una, es decir usando la opción **Evaluar celda(s)**:



Pantalla de wxMaxima mostrando la ventana con las opciones de 'Celda'

```
(%i1) k1:0.85; k2:0.12; y0:1;
(%o1) 0.85
(%o2) 0.12
(%o3) 1
Ecuación del modelo:
(%i4) ecuacion:'diff(y,t)=k1*y-k2*y;
```

(%o4) $\frac{d}{d\,t}\,y=0.73\,y$

```
(%i5) ode2(ecuacion,y,t);
rat: replaced -0.73 by -73/100 = -0.73
rat: replaced -0.73 by -73/100 = -0.73
rat: replaced -0.73 by -73/100 = -0.73
```

(%o5) $y=\%c\,\%e^{\frac{73\,t}{100}}$

```
(%i6) ic1(ode2(ecuacion,y,t),t=0,y=y0);
rat: replaced -0.73 by -73/100 = -0.73
rat: replaced -0.73 by -73/100 = -0.73
rat: replaced -0.73 by -73/100 = -0.73
```

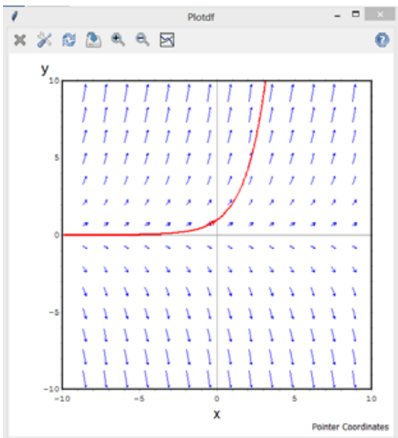
(%o6) $y=\%e^{\frac{73\,t}{100}}$

B. Campo de direcciones.

El estudio del campo de direcciones o pendientes requiere que carguemos **plotdf** con la orden **load**. Seguidamente en la orden **plotdf** especificaremos la expresión de la EDO e indicaremos que deseamos obtener la trayectoria que pasa por el punto de condiciones iniciales, tal y como se muestra a continuación:

```
--> load(plotdf);
--> plotdf(k1*y-k2*y,[trajectory_at,0,y0]);
```

De la evaluación de las dos celdas anteriores resultará la gráfica del campo de direcciones y la trayectoria de la *solución particular* que pasa por el punto (0, y_0):



En los fenómenos biológicos x es el *tiempo* **t**, razonándose a partir de la gráfica el **significado biológico** de la solución particular.

C. Curvas solución analítica y numérica: experimentos de simulación.

Además de la **solución analítica** (obtenida "a mano" o con wxMaxima usando las órdenes ode2 e ic1), y la **gráfica de la solución particular** (obtenida con wxMaxima con la orden plotdf), podemos realizar **experimentos de simulación** obteniéndose en cada experimento la **solución numérica**. Un experimento de simulación consiste en hacer una predicción del estado o comportamiento futuros del sistema o fenómeno cambiando en cada experimento el valor (i) de los parámetros, (ii) las condiciones iniciales o (iii) ambos a la vez, es decir de los parámetros y condiciones iniciales. La curva solución numérica se obtiene aplicando el *método numérico de Runge-Kutta de 4 orden*, utilizándose para tal fin la orden **rk**. Asignamos un nombre al procedimiento rk, por ejemplo **numsolrk**, el cual recogerá una lista de todos los puntos (t, y) de la curva, siendo necesario que especifiquemos la siguiente información:

rk(expresión de la EDO, variable en estudio (y), valor inicial de la variable en estudio (y_0), [variable independiente (t), valor inicial de t (0), valor final de t (por ej. 10), precisión o intervalo de integración (0.1, 0.01, 0.001, etc.)]);

```
-->  numsolrk:rk(k1*y-k2*y,y,y0,[t,0,10,0.01]);
-->  numsolrk;
```

Una vez que evaluemos las celdas, obtendremos los siguientes resultados:

```
(%i9) numsolrk:rk(k1·y-k2·y,y,y0,[t,0,10,0.01]);
(%o9) [[0,1],[0.01,1.007326709954493],[0.02,1.014707100587743],[0.03,1.022141565202513],[0.04,1.029630499983183],[0.05,1.037174304016859],[0.06,
1.044773379314644],[0.07,1.052428130833057],[0.08,1.06013896649562],[0.09,1.067906297214589],[0.1,1.075730536912856],[0.11,1.083612102546008],[0.12,
1.09155141412454],[0.13,1.099548894736247],[0.14,1.107604970568763],[0.15,1.115720070932274],[0.16,1.123894628282401],[0.17,1.132129078243239],[0.18,
1.140423859630574],[0.19,1.14877941447527],[0.2,1.157196188046822],[0.21,1.165674628877086],[0.22,1.174215188784179],[0.23,1.182818322896561],[0.24,
1.191484489677284],[0.25,1.200214150948426],[0.26,1.209007771915703],[0.27,1.217865821193257],[0.28,1.22678877082863],[0.29,1.23577709632792],[0.3,
1.244831276681119],[0.31,1.253951794387643],[0.32,1.263139135482037],[0.33,1.272393789559882],[0.34,1.281716249803885],[0.35,1.291107013010158],[0.36,
1.300566579614695],[0.37,1.310095453720038],[0.38,1.319694143122145],[0.39,1.329363159337443],[0.4,1.339103017630097],[0.41,1.348914237039459],[0.42,
1.358797340407732],[0.43,1.368752854407836],[0.44,1.378781309571466],[0.45,1.388883240317371],[0.46,1.399059184979833],[0.47,1.409309685837349],[0.48,
1.419635289141537],[0.49,1.430036545146239],[0.5,1.44051400813685],[0.51,1.451068236459853],[0.52,1.461699792552571],[0.53,1.472409242973146],[0.54,
1.483197158430724],[0.55,1.494064113815874],[0.56,1.505010688231219],[0.57,1.5160374650223],[0.58,1.527145031808663],[0.59,1.53833398051517],[0.6,
1.549604907403545],[0.61,1.560958413104149],[0.62,1.572395102647988],[0.63,1.583915585498954],[0.64,1.595520475586306],[0.65,1.607210391337381],[0.66,
1.618985955710556],[0.67,1.630847796228445],[0.68,1.642796545011334],[0.69,1.654832838810875],[0.7,1.666957319044012],[0.71,1.679170631827166],[0.72,
1.691473428010666],[0.73,1.703866363213431],[0.74,1.716350097857912],[0.75,1.728925297205282],[0.76,1.74159263139089],[0.77,1.754352775459973],[0.78,
1.767206409403628],[0.79,1.780154218195048],[0.8,1.79319689182603],[0.81,1.806335125343737],[0.82,1.819569618887743],[0.83,1.83290107772734],[0.84,
1.846330212299125],[0.85,1.859857738244858],[0.86,1.873484376449597],[0.87,1.887210853080116],[0.88,1.901037899623605],[0.89,1.914966252926645],[0.9,
1.92899665523448],[0.91,1.94312985423057],[0.92,1.957366603076433],[0.93,1.971707660451785],[0.94,1.986153790594966],[0.95,2.000705763343671],[0.96,
2.015364354175973],[0.97,2.030130344251643],[0.98,2.045004520453789],[0.99,2.059987675430781],[1.0,2.075080607638492],[1.01,2.090284121382851],[1.02,
2.105599026862705],[1.03,2.12102614021299],[1.04,2.136566283548228],[1.05,2.152220285006334],[1.06,2.167988978792751],[1.07,2.183873205224902],[1.08,
2.199873810776973],[1.09,2.21599164812502],[1.1,2.23222757619241],[1.11,2.248582460195592],[1.12,2.265057171690205],[1.13,2.281652588617523],[1.14,
2.29836959535124],[1.15,2.315209082744604],[1.16,2.33217194817788],[1.17,2.349259095606183],[1.18,2.366471435607644],[1.19,2.383809885431933],[1.2,
2.401275369049145],[1.21,2.418868817199035],[1.22,2.436591167440619],[1.23,2.454443364202135],[1.24,2.472426358831374],[1.25,2.490541109646373],[1.26,
2.508788581986492],[1.27,2.52716974826385],[1.28,2.545685588015148],[1.29,2.564337087953867],[1.3,2.583125242022853],[1.31,2.602051051447283],[1.32,
2.62111552478802],[1.33,2.64031967799536],[1.34,2.659664534463171],[1.35,2.679151125083434],[1.36,2.698780488301173],[1.37,2.7185536701698],[1.38,
2.738471724406856],[1.39,2.758535712450164],[1.4,2.778746703514396],[1.41,2.799105774648049],[1.42,2.819614010790841],[1.43,2.840272504831529],[1.44,
2.86108235766615],[1.45,2.882044678256686],[1.46,2.903160583690162],[1.47,2.924431199238175],[1.48,2.945857658416863],[1.49,2.967441103047304],[1.5,
2.989182683316372],[1.51,3.011083557838023],[1.52,3.033144893715044],[1.53,3.055367866601245],[1.54,3.077753660764109],[1.55,3.100303469147905],[1.56,
3.123018493437259],[1.57,3.145899944121191],[1.58,3.168949040557621],[1.59,3.192167011038355],[1.6,3.215555092854533],[1.61,3.239114532362569],[1.62,
3.262846585050572],[1.63,3.286752515605245],[1.64,3.310833597979284],[1.65,3.335091115459267],[1.66,3.359526360734043],[1.67,3.384140635963614],[1.68,
```

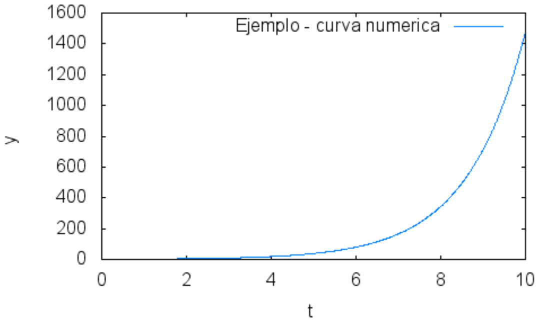
que vuelven de nuevo a mostrarse al invocar a **numsolrk**:

```
(%i10) numsolrk;
(%o10) [[0,1],[0.01,1.007326709954493],[0.02,1.014707100587743],[0.03,1.022141565202513],[0.04,1.029630499983183],[0.05,1.037174304016859],[0.06,
1.044773379314644],[0.07,1.052428130833057],[0.08,1.06013896649562],[0.09,1.067906297214589],[0.1,1.075730536912856],[0.11,1.083612102546008],[0.12,
1.09155141412454],[0.13,1.099548894736247],[0.14,1.107604970568763],[0.15,1.115720070932274],[0.16,1.123894628282401],[0.17,1.132129078243239],[0.18,
1.140423859630574],[0.19,1.14877941447527],[0.2,1.157196188046822],[0.21,1.165674628877086],[0.22,1.174215188784179],[0.23,1.182818322896561],[0.24,
1.191484489677284],[0.25,1.200214150948426],[0.26,1.209007771915703],[0.27,1.217865821193257],[0.28,1.22678877082863],[0.29,1.23577709632792],[0.3,
1.244831276681119],[0.31,1.253951794387643],[0.32,1.263139135482037],[0.33,1.272393789559882],[0.34,1.281716249803885],[0.35,1.291107013010158],[0.36,
1.300566579614695],[0.37,1.310095453720038],[0.38,1.319694143122145],[0.39,1.329363159337443],[0.4,1.339103017630097],[0.41,1.348914237039459],[0.42,
1.358797340407732],[0.43,1.368752854407836],[0.44,1.378781309571466],[0.45,1.388883240317371],[0.46,1.399059184979833],[0.47,1.409309685837349],[0.48,
1.419635289141537],[0.49,1.430036545146239],[0.5,1.44051400813685],[0.51,1.451068236459853],[0.52,1.461699792552571],[0.53,1.472409242973146],[0.54,
1.483197158430724],[0.55,1.494064113815874],[0.56,1.505010688231219],[0.57,1.5160374650223],[0.58,1.527145031808663],[0.59,1.53833398051517],[0.6,
1.549604907403545],[0.61,1.560958413104149],[0.62,1.572395102647988],[0.63,1.583915585498954],[0.64,1.595520475586306],[0.65,1.607210391337381],[0.66,
1.618985955710556],[0.67,1.630847796228445],[0.68,1.642796545011334],[0.69,1.654832838810875],[0.7,1.666957319044012],[0.71,1.679170631827166],[0.72,
```

El tiempo de espera o velocidad de cálculo necesario para que wxMaxima muestre los resultados depende del hardware del ordenador.

Finalmente, la representación gráfica de las *curvas solución* y *analítica* se obtiene con la orden **wxplot2d**. Por ejemplo, si deseamos obtener la **curva numérica**, escribiremos en wxMaxima y evaluaremos, obteniendo:

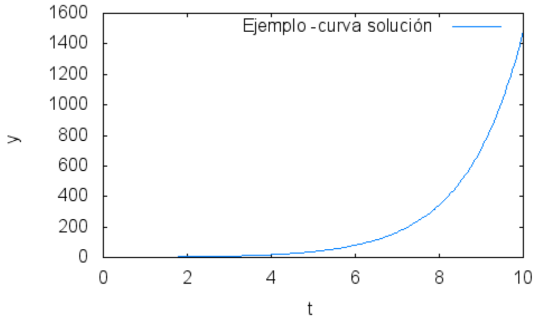
wxplot2d([discrete,numsolrk],[legend,"Ejemplo curva numerica"], [xlabel,"t"], [ylabel,"y"], [color,blue])$



De forma similar si lo que deseamos es obtener la **curva analítica**, escribiremos la orden:

wxplot2d(%e^(k1*t-k2*t),[t,0,10],[legend,"Ejemplo -curva solución "],[xlabel,"t"],[ylabel,"y"],[color,blue])$

resultando el siguiente gráfico:



Compruebe que en ambos casos la sintaxis de wxplot2d es similar, excepto al principio de la orden, utilizándose **([discrete,numsolrk]** en un caso, y **%e^(k1*t-k2*t),[t,0,10]** en el otro. Aunque obviamente ambas curvas deben ser similares, en la práctica utilizaremos la curva numérica para establecer las **predicciones** del fenómeno en estudio así como las correspondientes **conclusiones**.

I.2. Aplicación del modelo logístico a datos experimentales

En la práctica la aplicación del modelo logístico suele utilizar la expresión:

$$y = \frac{k}{1 + ce^{r(t-t_0)}}$$

en lugar de esta otra:

$$y = \frac{ky_0}{y_0 + (k - y_0)e^{-rt}}$$

que es la habitual en explicaciones teóricas.

Supóngase que estudiamos a lo largo de 22 días el crecimiento de un cultivo de levaduras en el laboratorio siendo el tope *k* igual a 668. Seguidamente, describiremos los pasos para la aplicación con wxMaxima del modelo logístico al experimento descrito.

A. Tabla de datos experimentales.

En primer lugar, indicaremos en que instante se define el **tiempo inicial** (que llamaremos por ej. T0) y el valor del tope *k* (668), construyendo con wxMaxima la **matriz** que representa a la **tabla de datos** del experimento:

```
-->  T0:0;
-->   K:668;
-->  t:[0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21];
-->  y:[10,18,30,47,72,120,180,256,352,442,514,560,592,631,638,652,654,662,664,665,667,665];
-->  tabladatos:transpose(matrix(t,y));
-->  length(tabladatos);
```

Observe que al definir los valores del tiempo *t* y del crecimiento de la levadura *y* como dos vectores fila, debemos trasponer la matriz resultante con la orden **transpose**:

```
(%i5) tabladatos:transpose(matrix(t,y));
```

De forma auxiliar se usa la orden **length** que devuelve el número de filas (número de observaciones o datos experimentales) de la matriz.

Al evaluar las celdas se obtiene finalmente la matriz de datos, y a la que de forma arbitraría hemos llamado **tabladatos**:

(%o5)
$$\begin{bmatrix} 0 & 10 \\ 1 & 18 \\ 2 & 30 \\ 3 & 47 \\ 4 & 72 \\ 5 & 120 \\ 6 & 180 \\ 7 & 256 \\ 8 & 352 \\ 9 & 442 \\ 10 & 514 \\ 11 & 560 \\ 12 & 592 \\ 13 & 631 \\ 14 & 638 \\ 15 & 652 \\ 16 & 654 \\ 17 & 662 \\ 18 & 664 \\ 19 & 665 \\ 20 & 667 \\ 21 & 665 \end{bmatrix}$$

```
(%i6) length(tabladatos);
(%o6) 22
```

B. Estimación de los parámetros c y r del modelo logístico.

En algunas ocasiones el tiempo inicial es distinto de 0 o se trata de una fecha como ocurre por ejemplo en estudios con poblaciones humanas, efectuándose en tal caso la siguiente operación:

```
--> T:col(tabladatos,1)-T0;
```

A continuación, y tras multiplicar por *y* el denominador de la expresión, reordenar los términos y tomar logaritmos se obtiene:

$$\ln\left(\frac{k}{y}-1\right)=\ln(c)+r(t-t_0)$$

o lo que es lo mismo:

$$\mathrm{YT} = \ln(c) + r\mathrm{T}$$

```
-->  YT:log(K/col(tabladatos,2)-1),numer;
```

En ambas tareas hemos utilizado la orden **col(tabladatos, num)** extrayendo en un caso el primer vector columna y en el otro el segundo vector columna de la matriz con nombre tabladatos. El parámetro **numer** le ordena a wxMaxima que exprese numéricamente los valores. Si evaluamos ambas celdas, obtendremos:

```
(%i7) T:col(tabladatos,1)-T0;
```

(%o7) $\begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \\ 16 \\ 17 \\ 18 \\ 19 \\ 20 \\ 21 \end{bmatrix}$

```
(%i8) YT:log(K/col(tabladatos,2)-1),numer;
```

$$
(\%o8)\quad \begin{bmatrix} 4.186619838331271 \\ 3.586600604993518 \\ 3.057140901682634 \\ 2.58118348022342 \\ 2.113574548049294 \\ 1.518783544165969 \\ 0.99735855496294 \\ 0.47584590486996 \\ -0.10788896201119 \\ -0.67077488280541 \\ -1.205270663041536 \\ -1.645805556604975 \\ -2.052773294597674 \\ -2.836387949896987 \\ -3.057140901682637 \\ -3.707455839686868 \\ -3.844050021841939 \\ -4.703506086708949 \\ -5.111987788356531 \\ -5.401174751987727 \\ -6.502790045915635 \\ -5.401174751987727 \end{bmatrix}
$$

Ahora construiremos con la orden **addcol** la matriz de datos transformados a la que ahora llamaremos **matrizdatos**:

```
--> matrizdatos:addcol(T,YT);
```

obteniendo tras la evaluación de la celda:

```
(%i9) matrizdatos:addcol(T,YT);
```

$$
(\%o9)\quad \begin{bmatrix} 0 & 4.186619838331271 \\ 1 & 3.586600604993518 \\ 2 & 3.057140901682634 \\ 3 & 2.58118348022342 \\ 4 & 2.113574548049294 \\ 5 & 1.518783544165969 \\ 6 & 0.99735855496294 \\ 7 & 0.47584590486996 \\ 8 & -0.10788896201119 \\ 9 & -0.67077488280541 \\ 10 & -1.205270663041536 \\ 11 & -1.645805556604975 \\ 12 & -2.052773294597674 \\ 13 & -2.836387949896987 \\ 14 & -3.057140901682637 \\ 15 & -3.707455839686868 \\ 16 & -3.844050021841939 \\ 17 & -4.703506086708949 \\ 18 & -5.111987788356531 \\ 19 & -5.401174751987727 \\ 20 & -6.502790045915635 \\ 21 & -5.401174751987727 \end{bmatrix}
$$

Seguidamente estimaremos los parámetros del modelo logístico por medio del **método de los mínimos cuadrados** (**Caja 1**). El método se aplica a los puntos (T, YT) obteniéndose la **recta de regresión lineal** YT = a + b T como sigue. En primer lugar, se carga la biblioteca de órdenes estadísticas **stats**:

```
--> load(stats);
```

En segundo lugar, se escribe la llamada al método de regresión lineal:

```
--> simple_linear_regression(matrizdatos);
```

Caja 1

Supóngase que realizamos un cierto experimento, en donde tomamos los valores de dos medidas X_i e Y_i en *n* individuos extraídos al azar. Si a continuación representamos gráficamente (X_1, Y_1), (X_2, Y_2), …, (X_n, Y_n) obtendremos una nube de puntos. Si la distribución de los puntos muestra una tendencia lineal entonces podremos estimar los coeficientes o parámetros *a* (origen en ordenadas) y *b* (pendiente) de la ecuación a partir de los **datos experimentales** (X_i, Y_i):

$$Y = a + b\,X$$

Si la distancia de los puntos (X_i, Y_i) a la recta es *mínima* entonces la recta se llama **recta de regresión**, y el procedimiento que permite estimar los coeficientes de la recta de regresión se conoce con el nombre de **método de los mínimos cuadrados** [7]:

$$b = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2}$$

$$a = \frac{\sum_i y_i - b\sum_i x_i}{n}$$

Si a continuación evaluamos las dos celdas, obtendremos:

```
(%i10) load(stats);
(%o10) C:/PROGRA~2/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/stats/stats.mac
(%i11) simple_linear_regression(matrizdatos);
                SIMPLE LINEAR REGRESSION
          model=3.987406850179485-0.49980097326318 x
                 correlation=-0.99578120010193
                 v_estimation=0.093913229107175
(%o11)  b_conf_int=[-0.52128302696656,-0.47831891955981]
                 hypotheses=H0: b = 0 ,H1: b # 0
                  statistic=48.53197815863209
                  distribution=[student_t,20]
                        p_value=0.0
```

En los resultados de wxMaxima que se muestran arriba leeremos la expresión de la recta de regresión:

model* =3.9874 – 0.4998 *x

(NOTA: tomamos cuatro decimales, ***model*** es YT y ***x*** es T)

estimando los parámetros del modelo, tal y como se muestra a continuación:

c = exp(a)   r= - b

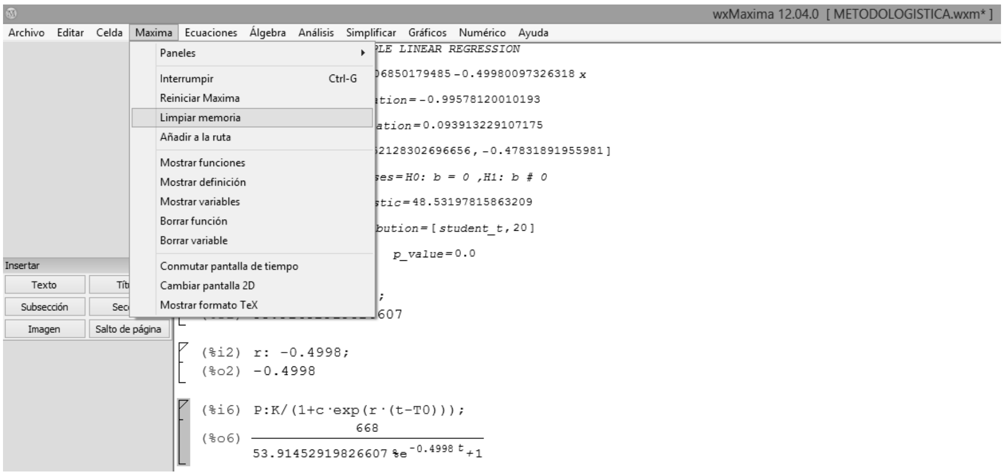
Utilizando de nuevo wxMaxima y evaluando las celdas, tendremos que:

```
(%i30) c: exp(3.9874);
(%o30) 53.91452919826607
(%i31) r: -0.4998;
(%o31) -0.4998
```

C. Curva del modelo y datos experimentales.

¿Cómo escribir la expresión del modelo? Se puede optar por hacerlo “a mano” o siguiendo el siguiente procedimiento.

En primer lugar, **Reiniciar Maxima** y **Limpiar memoria**, según se muestra a continuación:



Pantalla de wxMaxima mostrando las opciones de ‘Maxima’

En segundo lugar, si escribimos la celda %i6, y según el ejemplo, evaluamos una a una, y por este orden, %i1, %i2, %i30, %i31 y %i6 entonces *se escribirá* la expresión del modelo:

```
(%i6) P:K/(1+c*exp(r*(t-T0)));
```

$$(\%o6) \quad \frac{668}{53.91452919826607\,\%e^{-0.4998\,t}+1}$$

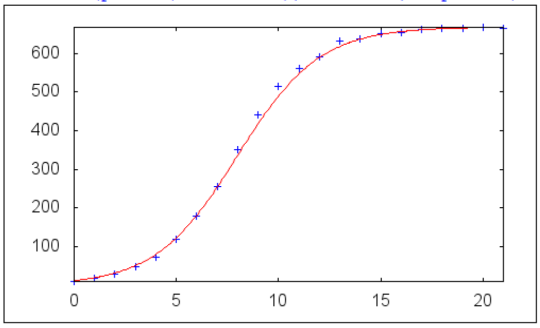
Finalmente, escribiremos la orden **wxdraw2d**:

wxdraw2d(points(tabladatos),color=red, explicit(668/(53.91452919826607*%e^(-0.4998*x)+1),x,0,21));

y una vez evaluada la celda obtendremos con wxMaxima - en color rojo - la **curva de la solución particular del modelo**, es decir la curva de la expresión %o6, y en color azul los **puntos** que representan los **datos experimentales**:

```
(%i15) wxdraw2d(points(tabladatos),color=red, explicit(668/(53.91452919826607*%e^(-0.4998*x)+1),x,0,21));
```

(%t15)



(%o15)

A partir de la representación gráfica estaremos en condiciones de preguntarnos ¿se ajustan los datos experimentales al modelo? Obviamente sí, un simple vistazo nos informa de la **bondad del modelo**.

I.3. Aplicación del modelo de von Bertalanffy a datos experimentales

Supóngase que estudiamos durante 6 meses el crecimiento (cm) de una especie de peces con interés en acuicultura, siendo su tamaño inicial (L0) igual a 9.6 cm. Seguidamente, describiremos los pasos para la aplicación con wxMaxima del modelo de von Bertalanffy al experimento descrito.

A. Tabla de datos experimentales.

En primer lugar, especificaremos el tamaño inicial de los peces (L0) para el **tiempo inicial** *t*=0, construyendo con wxMaxima la **matriz** que representa a la **tabla de datos** del experimento:

```
--> L0:9.6;
--> t:[1,2,3,4,5,6];
--> y:[22.8,33.6,38.1,44.0,47.7,51.2];
--> tabladatos:transpose(matrix(t,y));
--> long:length(tabladatos);
```

Una vez evaluadas las celdas, obtendremos:

```
(%i1) L0:9.6;
(%o1) 9.6
(%i2) t:[1,2,3,4,5,6];
(%o2) [1,2,3,4,5,6]
(%i3) y:[22.8,33.6,38.1,44.0,47.7,51.2];
(%o3) [22.8,33.6,38.1,44.0,47.7,51.2]
```

```
(%i4) tabladatos:transpose(matrix(t,y));
```

$$(\%o4) \begin{bmatrix} 1 & 22.8 \\ 2 & 33.6 \\ 3 & 38.1 \\ 4 & 44.0 \\ 5 & 47.7 \\ 6 & 51.2 \end{bmatrix}$$

```
(%i5) long:length(tabladatos);
(%o5) 6
```

B. Estimación de los parámetros L_{max} y k de von Bertalanffy.

Asumiremos que la solución particular de la EDO de von Bertalanffy es:

$$L = L_{\max} + (L_0 - L_{\max})e^{-kt}$$

Por consiguiente, deberemos estimar los valores de la **talla** o **longitud máxima** (L_{max}) para esa especie y el **parámetro de curvatura** (*k*). El procedimiento que utilizaremos representa los puntos formados por parejas de valores de la talla medida en el tiempo *t* y *t*+1, es decir (L_t, L_{t+1}). En el ejemplo llamaremos **matrizdatos** a la matriz formada por estos puntos, y que obtendremos con wxMaxima según se muestra a continuación:

```
-->  a[i,j]:=tabladatos[i+1,j];
-->  genmatrix(a,5,2);
-->  LY:col(%,2);
-->  LX:col(tabladatos,2);
-->  LX:submatrix(long,LX);
-->  LX:list_matrix_entries(LX);
-->  LY:list_matrix_entries(LY);
-->  matrizdatos:transpose(matrix(LX,LY));
```

Una vez evaluadas las celdas:

```
(%i12) a[i,j]:=tabladatos[i+1,j];
```

$$(\%o12)\ a_{i,j} := tabladatos_{i+1,j}$$

```
(%i13) genmatrix(a,5,2);
```

$$(\%o13) \begin{bmatrix} 2 & 33.6 \\ 3 & 38.1 \\ 4 & 44.0 \\ 5 & 47.7 \\ 6 & 51.2 \end{bmatrix}$$

```
(%i14) LY:col(%,2);
```

$$(\%o14) \begin{bmatrix} 33.6 \\ 38.1 \\ 44.0 \\ 47.7 \\ 51.2 \end{bmatrix}$$

```
(%i15) LX:col(tabladatos,2);
```

$$(\%o15) \begin{bmatrix} 22.8 \\ 33.6 \\ 38.1 \\ 44.0 \\ 47.7 \\ 51.2 \end{bmatrix}$$

```
(%i16) LX:submatrix(long,LX);
```

$$(\%o16) \begin{bmatrix} 22.8 \\ 33.6 \\ 38.1 \\ 44.0 \\ 47.7 \end{bmatrix}$$

```
(%i17) LX:list_matrix_entries(LX);
(%o17) [22.8,33.6,38.1,44.0,47.7]
(%i18) LY:list_matrix_entries(LY);
(%o18) [33.6,38.1,44.0,47.7,51.2]
(%i19) matrizdatos:transpose(matrix(LX,LY));
```

$$(\%o19) \begin{bmatrix} 22.8 & 33.6 \\ 33.6 & 38.1 \\ 38.1 & 44.0 \\ 44.0 & 47.7 \\ 47.7 & 51.2 \end{bmatrix}$$

La recta de regresión:

$$L_{t+1} = a + bL_t$$

se obtiene aplicando el método de los mínimos cuadrados (**véase Caja 1**). Con este fin cargamos con la orden **load** la biblioteca **stats**, obteniendo la recta de regresión con la orden **simple_linear_regression** cuya entrada es la matriz de nombre **matrizdatos**. Los parámetros de von Bertalanffy son estimados a partir de las siguientes expresiones:

$$L_{max} = \frac{a}{1-b} \qquad k = -\ln(b)$$

Realizando estos pasos con wxMaxima, y una vez evaluadas las celdas, se obtienen los resultados que se muestran más abajo. Observe que en wxMaxima la orden **log(x)** devuelve el logaritmo neperiano de x, y que es suficiente con tomar cuatro decimales de los valores a y b de la recta de regresión.

```
(%i20) load(stats);
(%o20) C:/PROGRA~2/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/stats/stats.mac
(%i21) simple_linear_regression(matrizdatos);
                 SIMPLE LINEAR REGRESSION
          model=0.72025253434742 x+16.09779562090206
                correlation=0.9831393925175
              v_estimation=2.258865572774871
(%o21)  b_conf_int=[0.47411186087788,0.96639320781696]
              hypotheses=H0: b = 0 ,H1: b # 0
                statistic=9.312418725825726
                distribution=[student_t,3]
                 p_value=0.0026214541112177
(%i22) Lmax:16.0977/(1-0.7202);
(%o22) 57.53288062902072
(%i23) K:-log(0.7202);
(%o23) 0.32822632776736
```

C. Curva del modelo y datos experimentales.

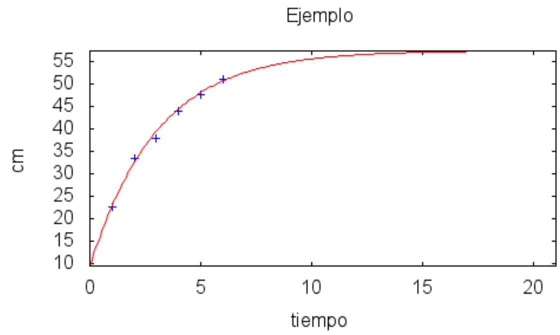
Finalmente, y concluidas las etapas anteriores, representaremos la curva del modelo siguiendo el procedimiento descrito anteriormente:

```
(%i4) L:Lmax+(L0-Lmax)*exp(-K*t);
(%o4) 57.53288062902072-47.93288062902072 %e^(-0.32822632776736 t)
```

y utilizando la orden **draw2d** (NOTA: escríbase en una sola línea):

wxdraw2d(points(tabladatos),color=red,explicit(57.53288062902072-47.93288062902072*%e^(-0.32822632776736*x), x,0,21),title="Ejemplo",xlabel="tiempo",ylabel="cm");

obtendremos la siguiente figura:



Recuérdese que en color rojo representamos la **curva de la solución particular del modelo**, es decir la que tiene la expresión %o4, y en azul los **datos experimentales**.

I.4. Aplicación del modelo de Leslie

Aunque *no se trate de una EDO* el modelo de Leslie permite **proyectar**, es decir predecir para *t*+1, la estructura de una población por clases de edad, incluyéndolo en este manual por su utilidad práctica en la gestión de poblaciones en la Naturaleza. Además, es un modelo que ilustra la utilidad del **álgebra lineal** en la simulación de los sistemas dinámicos. A diferencia de las EDOS la predicción es el resultado de multiplicar la matriz de Leslie por un vector de población en el tiempo *t*:

$$\begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_i \end{pmatrix}_{t+1} = \begin{pmatrix} f_1 & f_2 & \cdots & \cdots & f_i \\ s_1 & 0 & \cdots & \cdots & 0 \\ 0 & s_2 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & s_i \end{pmatrix} \begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_i \end{pmatrix}_{t}$$

siendo f_i el promedio de hijas por hembra en una cierta clase de edad *i* y s_i la tasa o probabilidad de que las hembras de la clase de edad *i* pasen, es decir sobrevivan, a la clase de edad *i*+1.

Supóngase que estudiamos una población de cabra montesa *Capra pyrenaica* obteniéndose los siguientes datos:

Clase de edad (i)	Censo (C_i)	Fecundidad (m_i)
0	1000	0.29 (no se utiliza)
1	498	0.70
2	317	1.1
3	149	1.1
4	51	1.1

A continuación, se muestran los pasos a seguir para estudiar esta población con wxMaxima:

A. Matriz de Leslie.

El modelo utiliza una matriz, la llamada **matriz de Leslie**, dándose en la práctica dos situaciones posibles: la *matriz es conocida* o hay que obtenerla partir de los *datos experimentales*. Consideraremos la segunda situación. En tal caso los valores de f_i y s_i son obtenidos como sigue (**método pre-nacimiento**):

$$S_i = \frac{C_{i+1}}{C_i} \qquad f_i = m_i \frac{C_1}{C_0}$$

Por consiguiente, utilizando los datos experimentales de que disponemos tendremos que:

s_1=317/498=0.636 s_2=149/317=0.470 s_3=51/149=0.342

siendo:

f_1= (0.7).(0.498) = 0.348

f_2=f_3=f_4= (1.1).(0.498) = 0.548

Si asumimos que los datos experimentales son tomados en el tiempo inicial t=0, entonces el **vector de población inicial** es:

```
--> N1[0]:498;
--> N2[0]:317;
--> N3[0]:149;
--> N4[0]:51;
```

tal que al evaluar las celdas se mostrará el contenido del vector:

```
(%i8) N1[0]:498;
(%o8) 498
(%i9) N2[0]:317;
(%o9) 317
(%i10) N3[0]:149;
(%o10) 149
(%i11) N4[0]:51;
(%o11) 51
```

A continuación, construiremos la matriz de Leslie a la que llamaremos **leslie**, obteniendo una vez evaluada la celda:

```
(%i12) leslie:matrix([0.348,0.548,0.548,0.548],[0.636,0,0,0],[0,0.470,0,0],[0,0,0.342,0]);
```

$$(\%o12)\quad \begin{bmatrix} 0.348 & 0.548 & 0.548 & 0.548 \\ 0.636 & 0 & 0 & 0 \\ 0 & 0.47 & 0 & 0 \\ 0 & 0 & 0.342 & 0 \end{bmatrix}$$

B. Proyección de la estructura de la población.

En primer lugar, construimos el vector de población inicial y definimos el tiempo máximo de simulación (Tmax), por ejemplo deseamos hacer una proyección o predicción a 15 años:

```
(%i8) N[0]:transpose([N1[0],N2[0],N3[0],N4[0]]);
```

$$(\%o8)\quad \begin{bmatrix} 498 \\ 317 \\ 149 \\ 51 \end{bmatrix}$$

```
(%i9) Tmax:15;
(%o9) 15
```

En segundo lugar, escribimos la siguiente rutina en lenguaje wxMaxima:

```
(%i10) for t:0 thru Tmax do block(
       N[t+1]:leslie.N[t],
       N[t]:N[t+1],
       print(t,N[t])
       );
```

Una vez evaluada la rutina, obtendremos en un listado el vector de población que para cada *t* (en el ejemplo, años) recoge la estructura de la población para las clases de edad 1, 2, 3 y 4:

$$\begin{bmatrix} 0 & \begin{bmatrix} 456.62 \\ 316.728 \\ 148.99 \\ 50.95800000000001 \end{bmatrix} \\ 1 & \begin{bmatrix} 442.042208 \\ 290.41032 \\ 148.86216 \\ 50.95458 \end{bmatrix} \\ 2 & \begin{bmatrix} 422.475117264 \\ 281.138844288 \\ 136.4928504 \\ 50.91085872 \end{bmatrix} \end{bmatrix}$$

$$
\begin{array}{cc}
3 & \begin{bmatrix} 403.782660075456 \\ 268.694174579904 \\ 132.13525681536 \\ 46.6805548368 \end{bmatrix} \\
4 & \begin{bmatrix} 385.7518381614298 \\ 256.80577180799 \\ 126.2862620525549 \\ 45.19025783085311 \end{bmatrix} \\
5 & \begin{bmatrix} 368.9403355270636 \\ 245.3381690706693 \\ 120.6987127497553 \\ 43.18990162197378 \end{bmatrix} \\
6 & \begin{bmatrix} 352.6475140898525 \\ 234.6460533952125 \\ 115.3089394632146 \\ 41.27895976041631 \end{bmatrix}
\end{array}
$$

…

$$
\begin{array}{cc}
12 & \begin{bmatrix} 269.1422028477761 \\ 179.0602350326687 \\ 88.03526409457841 \\ 31.49527039559492 \end{bmatrix} \\
13 & \begin{bmatrix} 257.2892282895435 \\ 171.1744410111856 \\ 84.15831046535428 \\ 30.10806032034582 \end{bmatrix} \\
14 & \begin{bmatrix} 245.9582163094545 \\ 163.6359491921497 \\ 80.45198727525721 \\ 28.78214217915117 \end{bmatrix} \\
15 & \begin{bmatrix} 235.126262374004 \\ 156.4294255728131 \\ 76.90889612031036 \\ 27.51457964813797 \end{bmatrix}
\end{array}
$$

(%o9) *done*

Es importante recordar que la computación en wxMaxima con vectores y matrices requiere **Reiniciar Maxima** y **Limpiar memoria**. También deberían borrarse todos los resultados previos.

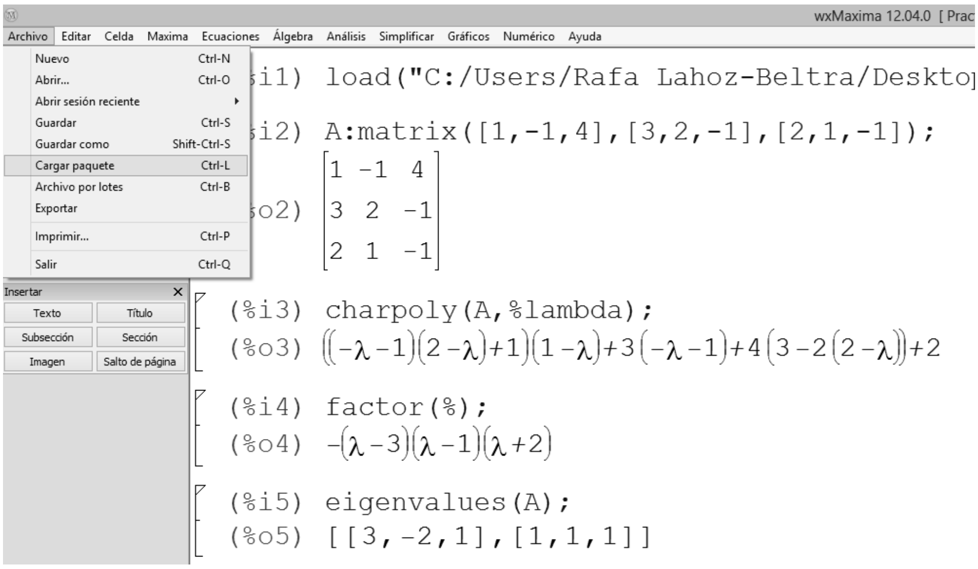
II. Métodos para la resolución con wxMaxima de sistemas de ecuaciones diferenciales lineales de primer orden con coeficientes constantes (SEDLPOH)

Se supone que un biólogo desea estudiar un fenómeno o sistema que puede ser representado por **compartimentos**, tal que a un compartimento *i* (población, tanque, órgano, sustancia química, etc.) le hacemos corresponder una cierta variable y_i que especifica su **estado**. Sabemos que la **tasa de cambio** y' puede ser expresada como la diferencia E – S, siendo E y S los términos de entrada y salida del compartimento respectivamente. La E recoge todo aquello que hace aumentar y la S disminuir la tasa de cambio (o derivada) del sistema o fenómeno que es objeto de estudio.

Sea un fenómeno o sistema biológico que deseamos estudiar y al que representaremos por el siguiente sistema de ecuaciones diferenciales (ED):

$$Y' = A.Y + F$$

A continuación, y según sea la **matriz de coeficientes A**, explicaremos los pasos a seguir para estudiar dicho fenómeno o sistema con wxMaxima. El estudio consiste en **resolver** el sistema de ecuaciones diferenciales lineales de primer orden **homogéneo** (abreviadamente **SEDLPOH)** y **predecir** el comportamiento futuro del fenómeno a partir de las **condiciones iniciales Y0**. Por consiguiente al tratarse de sistemas homogéneos sólo se estudian sistemas con el **término F** igual a 0, es decir que carecen de un término del tipo *t+1*, *exp(2t)*, *Sen(3t)*, etc. en la expresión ecuación diferencial.



Diagonalización de una matriz con wxMaxima, mostrando la opción de 'Cargar paquete'

II.1. A_{2x2} y valores propios reales con multiplicidad uno

Sea el sistema el que se muestra a continuación:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 5 & 4 \\ 1 & 5 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\begin{cases} y_1 = 5y_1 + 4y_2 \\ y_2 = y_1 + 5y_{23} \end{cases}$$

En primer lugar cargaremos el paquete **SEDLPOH.mac** (**véase APENDICE**), introducimos la matriz de coeficientes A y **diagonalizamos** obteniendo los **valores propios**, su **multiplicidad algebraica** y los **vectores propios**. La librería SEDLPOH.mac es una colección de rutinas con las que resolver sistemas de dos y tres ecuaciones con **valores propios reales**:

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$

(%i2) A:matrix([5,4],[1,5]);
```

(%o2) $\begin{bmatrix} 5 & 4 \\ 1 & 5 \end{bmatrix}$

```
(%i3) charpoly(A,%lambda);
```

(%o3) $(5-\lambda)^2 - 4$

```
(%i4) factor(%);
```

(%o4) $(\lambda - 7)(\lambda - 3)$

```
(%i5) eigenvalues(A);
```

(%o5) [[3,7],[1,1]]

```
(%i6) eigenvectors(A);
```

obteniéndose:

(%o6) $[[[3,7],[1,1]],[[[1,-\frac{1}{2}]],[[1,\frac{1}{2}]]]]$

En segundo lugar, introducimos los vectores propios **v1** y **v2**:

```
(%i7) v1:[2,-1];
```

$$(\%o7)\quad [2,-1]$$

```
(%i8) v1:transpose(v1);
```

$$(\%o8)\quad \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

```
(%i9) v2:[2,1];
```

$$(\%o9)\quad [2,1]$$

```
(%i10) v2:transpose(v2);
```

$$(\%o10)\quad \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

En tercer lugar, estaremos en condiciones de declarar las soluciones parciales del sistema **z1** y **z2**, utilizando para tal fin la orden **spah(lambda, v)**, siendo **lambda** el valor propio y **v** el vector propio asociado:

```
(%i11) z1:spah(3,v1);
```

$$(\%o11)\quad \begin{bmatrix} 2\,\%e^{3\,t} \\ -\%e^{3\,t} \end{bmatrix}$$

```
(%i12) z2:spah(7,v2);
```

$$(\%o12)\quad \begin{bmatrix} 2\,\%e^{7\,t} \\ \%e^{7\,t} \end{bmatrix}$$

Un paso importante consiste en verificar si la solución obtenida es válida, es decir una vez obtenida la **matriz fundamental G** ¿son sus vectores linealmente independientes? Con objeto de responder a esta pregunta obtendremos el **rango** de la matriz G. Adicionalmente, se puede calcular su

determinante que deberá ser distinto de cero. La matriz G se construye con la orden **addcol** siendo sus columnas son **z1** y **z2**:

```
(%i13) G:addcol(z1,z2);
```

$$(\%o13)\quad \begin{bmatrix} 2\,\%e^{3\,t} & 2\,\%e^{7\,t} \\ -\%e^{3\,t} & \%e^{7\,t} \end{bmatrix}$$

```
(%i14) rank(G);
```

(%o14) 2

```
(%i15) determinant(G);
```

$$(\%o15)\quad 4\,\%e^{10\,t}$$

En cuarto lugar, obtendremos la **solución general de la homogénea** (SGH), es decir Y(t)= G.C, tal que C es el vector de constantes $\begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ utilizando la orden **sghec2(G)** al tratarse de un sistema con dos ecuaciones:

```
(%i16) sghec2(G);
```

$$(\%o16)\quad \begin{bmatrix} 2\,c2\,\%e^{7\,t}+2\,c1\,\%e^{3\,t} \\ c2\,\%e^{7\,t}-c1\,\%e^{3\,t} \end{bmatrix}$$

En quinto, y último lugar, obtendremos la **solución particular de la homogénea** (SPH). Introducimos el vector de condiciones iniciales **Y0**, es decir los **datos experimentales** para el t=0. Seguidamente, con la orden **sisconstec2(Y0)** que resuelve el sistema de ecuaciones de las constantes c_1 y c_2 -que es compatible determinado- obtenemos sus valores. A continuación, con la orden **sphec2(**c_1**,** c_2**)** obtendremos la solución particular:

```
(%i17) Y0:[1,0];
```

(%o17) [1, 0]

(%i18) Y0:transpose(Y0);

(%o18) $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$

(%i19) sisconstec2(Y0);

(%o19) $[c1 = \frac{1}{4}, c2 = \frac{1}{4}]$

(%i20) sphec2(1/4,1/4);

(%o20) $\begin{bmatrix} \frac{\%e^{7t}}{2} + \frac{\%e^{3t}}{2} \\ \frac{\%e^{7t}}{4} - \frac{\%e^{3t}}{4} \end{bmatrix}$

Concluiremos representando gráficamente las soluciones particulares. Extraemos las expresiones de %o20 con **list_matrix_entries** almacenándolas en una variable a la que llamamos por ejemplo **solpal**, utilizando la orden **wxplot2d**:

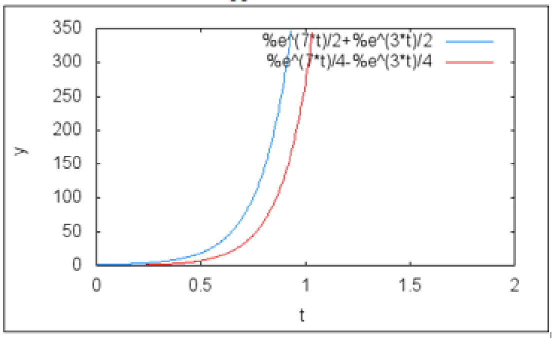
(%i21) solpal:list_matrix_entries(%);

(%o21) $[\frac{\%e^{7t}}{2} + \frac{\%e^{3t}}{2}, \frac{\%e^{7t}}{4} - \frac{\%e^{3t}}{4}]$

(%i22) wxplot2d(solpal,[t,0,2],[y,0,350]);

```
plot2d: some values were clipped.
plot2d: some values were clipped.
```

(%t22)



(%o22)

-->

Created with wxMaxima

II.2. A_{3x3} y valores propios reales con multiplicidad uno

Resolveremos el sistema que se muestra a continuación:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 10 \\ 5 \\ 2 \end{pmatrix}$$

$$\begin{cases} y_1 = y_1 - y_2 + 4y_3 \\ y_2 = 3y_1 + 2y_2 - y_3 \\ y_3 = 2y_1 + y_2 - y_3 \end{cases}$$

El método a seguir es similar al del ejemplo anterior, excepto que ahora el sistema es de tres ecuaciones y las órdenes específicas de la librería SEDLPOH.mac tienen ahora la terminación **ec3**. Sea el siguiente ejemplo, una vez cargada la librería, diagonalizamos la **matriz de coeficientes** con el fin de averiguar en qué caso estamos:

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$

(%i2) A:matrix([1,-1,4],[3,2,-1],[2,1,-1]);
```

$$(\%o2)\quad \begin{bmatrix} 1 & -1 & 4 \\ 3 & 2 & -1 \\ 2 & 1 & -1 \end{bmatrix}$$

```
(%i3) charpoly(A,%lambda);
```

$$(\%o3)\quad ((-\lambda-1)(2-\lambda)+1)(1-\lambda)+3(-\lambda-1)+4(3-2(2-\lambda))+2$$

```
(%i4) factor(%);
```

$$(\%o4)\quad -(\lambda-3)(\lambda-1)(\lambda+2)$$

```
(%i5) eigenvalues(A);

(%o5) [[3,-2,1],[1,1,1]]

(%i6) eigenvectors(A);

(%o6) [[[3,-2,1],[1,1,1]],[[[1,2,1]],[[1,-1,-1]],[[1,-4,-1]]]]
```

A continuación, introducimos los tres vectores propios **v1**, **v2** y **v3**, obteniendo las soluciones parciales correspondientes **z1**, **z2** y **z3**:

```
(%i7) v3:[1,2,1];

(%o7) [1,2,1]

(%i8) v3:transpose(v3);
```

$$(\%o8)\quad \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

```
(%i9) v2:[1,-1,-1];
```

$$(\%o9)\quad [1,-1,-1]$$

```
(%i10) v2:transpose(v2);
```

$$(\%o10)\quad \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$$

```
(%i11) v1:[1,-4,-1];
```

$$(\%o11)\quad [1,-4,-1]$$

```
(%i12) v1:transpose(v1);
```

$$(\%o12)\quad \begin{bmatrix} 1 \\ -4 \\ -1 \end{bmatrix}$$

```
(%i13) z1:spah(1,v1);
```

$$(\%o13)\quad \begin{bmatrix} \%e^{t} \\ -4\,\%e^{t} \\ -\%e^{t} \end{bmatrix}$$

```
(%i14) z2:spah(-2,v2);
```

$$(\%o14)\quad \begin{bmatrix} \%e^{-2\,t} \\ -\%e^{-2\,t} \\ -\%e^{-2\,t} \end{bmatrix}$$

```
(%i15) z3:spah(3,v3);
```

(%o15) $\begin{bmatrix} \%e^{3t} \\ 2\,\%e^{3t} \\ \%e^{3t} \end{bmatrix}$

Ahora es el momento de construir la **matriz fundamental G**, evaluando la validez de la solución obtenida:

```
(%i16) G:addcol(z1,z2,z3);
```

(%o16) $\begin{bmatrix} \%e^{t} & \%e^{-2t} & \%e^{3t} \\ -4\,\%e^{t} & -\%e^{-2t} & 2\,\%e^{3t} \\ -\%e^{t} & -\%e^{-2t} & \%e^{3t} \end{bmatrix}$

```
(%i17) rank(G);
```

(%o17) 3

```
(%i18) determinant(G);
```

(%o18) $6\,\%e^{2t}$

Puesto que el rango es 3, y el determinante distinto de cero, los vectores son linealmente independientes por lo que la solución encontrada es válida.

¿Cuál es la **solución general del sistema homogéneo**? La SGH, es decir Y(t)=G.C, se obtiene multiplicando la matriz G por el **vector de constantes** C, esto es el vector $\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$. Utilizamos ahora la orden **sghec3(G)**:

```
(%i19) sghec3(G);
```

(%o19) $\begin{bmatrix} c3\ \%e^{3\,t}+c1\ \%e^{t}+c2\ \%e^{-2\,t} \\ 2\ c3\ \%e^{3\,t}-4\ c1\ \%e^{t}-c2\ \%e^{-2\,t} \\ c3\ \%e^{3\,t}-c1\ \%e^{t}-c2\ \%e^{-2\,t} \end{bmatrix}$

Seguidamente, introducimos el **vector de condiciones iniciales Y0** con los **datos experimentales** en **t=0**, resolvemos el sistema de ecuaciones con **sisconstec3(Y0)** e introducimos los valores obtenidos de las constantes como parámetros de la orden **sphec3(c1, c2, c3)** cuya salida u output es la **solución particular del sistema homogéneo** (SPH):

(%i20) Y0:[10,5,2];

(%o20) [10, 5, 2]

(%i21) Y0:transpose(Y0);

(%o21) $\begin{bmatrix} 10 \\ 5 \\ 2 \end{bmatrix}$

(%i22) sisconstec3(Y0);

(%o22) $[c1=1, c2=3, c3=6]$

(%i23) sphec3(1,3,6);

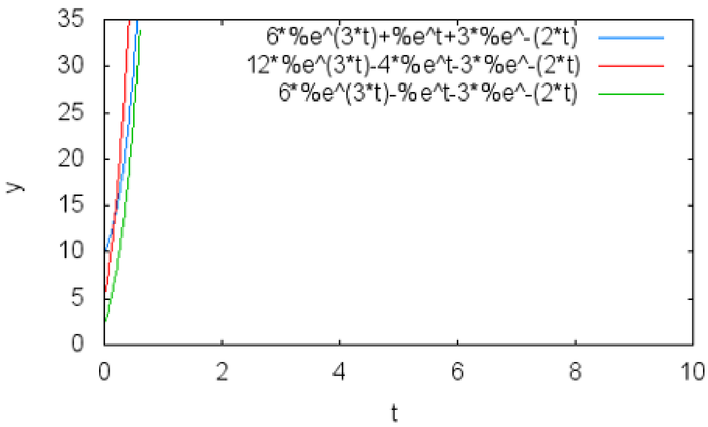
(%o23) $\begin{bmatrix} 6\ \%e^{3\,t}+\%e^{t}+3\ \%e^{-2\,t} \\ 12\ \%e^{3\,t}-4\ \%e^{t}-3\ \%e^{-2\,t} \\ 6\ \%e^{3\,t}-\%e^{t}-3\ \%e^{-2\,t} \end{bmatrix}$

Finalmente, representamos gráficamente las curvas solución particular aplicando el siguiente procedimiento. Definimos una variable a la que le damos un nombre arbitrario, por ej. **solpal** que recoge las expresiones vía **list_matrix_entries**, obteniendo las curvas con la orden **wxplot2d**:

(%i24) solpal:list_matrix_entries(%);

(%o24) $[6\,\%e^{3\,t}+\%e^{t}+3\,\%e^{-2\,t}, 12\,\%e^{3\,t}-4\,\%e^{t}-3\,\%e^{-2\,t}, 6\,\%e^{3\,t}-\%e^{t}-3\,\%e^{-2\,t}]$

(%i25) wxplot2d(solgnal,[t,0,10],[y,0,35]);



-->

Created with wxMaxima

II.3. A_{2x2} y valores propios complejos con multiplicidad uno

Sea el sistema a resolver:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 2 & 8 \\ -1 & -2 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 6 \\ 2 \end{pmatrix}$$

$$\begin{cases} y_1 = 2y_1 + 8y_2 \\ y_2 = -y_1 - 2y_2 \end{cases}$$

En primer lugar, y aunque en este caso no utilicemos ninguna de sus rutinas, cargamos siguiendo el protocolo de ejemplos anteriores el paquete SEDLPOH.mac; introducimos la matriz de coeficientes y diagonalizamos. Observamos que se trata de un **polinomio característico** p(λ)= $\lambda^2 + 4$ con raíces complejas: $\lambda_1 = 2i$ y $\lambda_1 = -2i$. Al tratarse de una **raíz compleja**, las soluciones parciales **z1** y **z2** las obtendremos de la "raíz positiva", es decir de 2*i* en el ejemplo.

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$

(%i2) A:matrix([2,8],[-1,-2]);
```

$$(\%o2) \begin{bmatrix} 2 & 8 \\ -1 & -2 \end{bmatrix}$$

```
(%i3) charpoly(A,%lambda);
```

$$(\%o3)\ (-\lambda-2)(2-\lambda)+8$$

```
(%i4) factor(%);
```

$$(\%o4)\ \lambda^2+4$$

```
(%i5) eigenvalues(A);

(%o5) [[-2 %i, 2 %i], [1, 1]]

(%i6) eigenvectors(A);
```

$$(\%o6)\ [[[-2\,\%i, 2\,\%i], [1, 1]], [[[1, -\frac{\%i+1}{4}]], [[1, \frac{\%i-1}{4}]]]]$$

Anotamos la parte real $\alpha = 0$ y la compleja $\beta = 2$ de la raíz del polinomio característico así como la parte real **Re** y compleja **Im** del vector propio complejo:

$$\begin{pmatrix} 1 \\ -\frac{1}{4}+\frac{1}{4}i \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{1}{4} \end{pmatrix} + i \begin{pmatrix} 0 \\ \frac{1}{4} \end{pmatrix}$$

y que es "razonado" por wxMaxima por un procedimiento similar al que se muestra en el recuadro inferior (**Caja 2**).

Caja 2

```
(%i10) ec1:8·w2+(2-2·%i)·w1;
(%o10) 8 w2+(2-2 %i) w1
(%i11) ec2:(-2·%i-2)·w2-w1;
(%o11) (-2 %i-2) w2-w1
(%i12) linsolve([ec1,ec2],[w1,w2]);
olve: dependent equations eliminated: (2)
(%o12) [w1=%r1, w2=(%i %r1-%r1)/4]
```

No obstante, y para ejercitarnos, utilizaremos en el ejemplo el vector propio obtenido "a mano":

$$\begin{pmatrix} 2+2i \\ -1 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix} + i \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

y que nosotros "razonamos" a partir de un procedimiento en wxMaxima, que mostramos en el siguiente recuadro (**Caja 3**).

Caja 3

```
(%i6) M:A-(2·%i).ident(2);
(%o6) [2-2 %i      8     ]
      [ -1     -2 %i-2   ]
(%i7) x: [w1,w2];
(%o7) [w1, w2]
(%i8) x: transpose(x);
(%o8) [w1]
      [w2]
(%i9) M.x;
(%o9) [8 w2+(2-2 %i) w1 ]
      [(-2 %i-2) w2-w1  ]
```

Obsérvese que M.x es equivalente a escribir M.x=0, obteniendo el vector propio que utilizaremos a partir de las expresiones en **%o9**, tal que -(2+2i)w2 = w1.

Volviendo al ejemplo que estamos resolviendo, continuaremos como se indica a continuación. Obviamente, si se prefiere, también podríamos haber utilizado el vector propio que se obtiene con wxMaxima:

(%i7) Re:transpose([2,-1]);

(%o7) $\begin{bmatrix} 2 \\ -1 \end{bmatrix}$

(%i8) Im:transpose([2,0]);

(%o8) $\begin{bmatrix} 2 \\ 0 \end{bmatrix}$

(%i9) v:(Re+Im*%i);

(%o9) $\begin{bmatrix} 2\,\%i+2 \\ -1 \end{bmatrix}$

(%i10) %alpha:0;

(%o10) 0

(%i11) %beta:2;

(%o11) 2

Concluida esta etapa, cargamos la librería **SEDLPOHRC.mac** (**véase APENDICE**) que es una colección de rutinas para sistemas A_{2x2} y A_{3x3} con raíces complejas. Las órdenes específicas de la librería SEDLPOHRC.mac tienen ahora la terminación **i**.

La orden **spahi**, y que ejecutaremos con **$** en lugar de **;** con el fin de evitar que nos muestre operaciones intermedias, proporciona las soluciones parciales **z1** y **z2**. Si quisiéramos obtenerlas "a mano" entonces tales soluciones se obtendrían a partir de las siguientes expresiones (**Caja 4**).

Caja 4

Sea el producto del vector propio complejo por $e^{\lambda t}$, tal que utilizando la expresión de Euler, obtendremos:

$$\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}.e^{\lambda t} = \left[\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + i\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right].e^{(\alpha+\beta)t} =$$

$$\left[\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + i\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right].e^{\alpha t}.e^{i\beta t} =$$

$$\left[\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} + i\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}\right].e^{\alpha t}.\left(\cos\beta t + i\, sen\beta t\right):$$

Por tanto, las soluciones parciales son:

$$Z_1(t) = e^{\alpha t}\left[\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}^{\mathrm{Re}} \cos\beta t - \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}^{\mathrm{Im}} sen\beta t\right]$$

$$Z_2(t) = e^{\alpha t}\left[\begin{pmatrix} w_1 \\ w_2 \end{pmatrix}^{\mathrm{Im}} \cos\beta t + \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}^{\mathrm{Re}} sen\beta t\right]$$

Con wxMaxima tendremos que:

```
(%i12) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOHRC.mac")$

(%i13) spahi$
```

(%i14) z1;

$$(\%o14)\quad \begin{bmatrix} 2\cos(2\,t)-2\sin(2\,t) \\ -\cos(2\,t) \end{bmatrix}$$

(%i15) z2;

$$(\%o15)\quad \begin{bmatrix} 2\sin(2\,t)+2\cos(2\,t) \\ -\sin(2\,t) \end{bmatrix}$$

Construiremos ahora la **matriz fundamental** G:

(%i16) G:addcol(z1,z2);

$$(\%o16)\quad \begin{bmatrix} 2\cos(2\,t)-2\sin(2\,t) & 2\sin(2\,t)+2\cos(2\,t) \\ -\cos(2\,t) & -\sin(2\,t) \end{bmatrix}$$

obteniendo la solución general del sistema homogéneo (SGH) con la orden **sghi**:

(%i17) sghi;

$$(\%o17)\quad \begin{bmatrix} c2\,(2\sin(2\,t)+2\cos(2\,t))+c1\,(2\cos(2\,t)-2\sin(2\,t)) \\ -c2\sin(2\,t)-c1\cos(2\,t) \end{bmatrix}$$

Aunque para esta clase de sistema G es siempre válida, siendo sus vectores linealmente independientes, se puede comprobar, confirmándose su validez:

(%i18) rank(G);

(%o18) 2

```
(%i19) determinant(G);
```

$$(\%o19)\quad \cos(2\,t)(2\sin(2\,t)+2\cos(2\,t))-(2\cos(2\,t)-2\sin(2\,t))\sin(2\,t)$$

Introduciremos el **vector de condiciones iniciales Y0**, resolvemos el sistema de constantes con la orden **sisconsti(Y0)**, obteniendo finalmente las **soluciones particulares del sistema homogéneo** (SPH) con la orden **sphi(c1, c2)**:

```
(%i20) Y0:transpose([6,4]);
```

$$(\%o20)\quad \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

```
(%i21) sisconsti(Y0);
```

$$(\%o21)\quad [c1=-4,\ c2=7]$$

```
(%i22) sphi(-4,7);
```

$$(\%o22)\quad \begin{bmatrix} 7\,(2\sin(2\,t)+2\cos(2\,t))-4\,(2\cos(2\,t)-2\sin(2\,t)) \\ 4\cos(2\,t)-7\sin(2\,t) \end{bmatrix}$$

Las curvas de la solución particular se pueden representar gráficamente aplicando el protocolo descrito anteriormente:

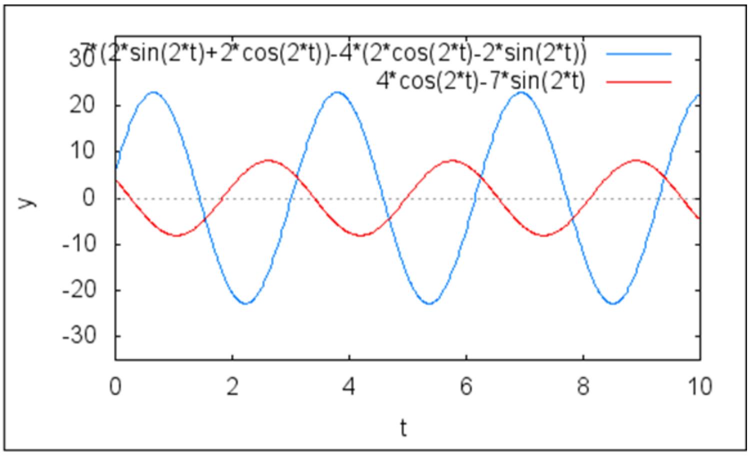
```
(%i23) solpal:list_matrix_entries(%);
```

$$(\%o23)\quad [7\,(2\sin(2\,t)+2\cos(2\,t))-4\,(2\cos(2\,t)-2\sin(2\,t)),\ 4\cos(2\,t)-7\sin(2\,t)]$$

```
(%i24) wxplot2d(solpal,[t,0,10],[y,-35,35]);
```

Aunque sea un **sistema lineal** ¿A qué modelo *no lineal* recuerdan las curvas obtenidas? A simple vista, y sin entrar cuestiones teóricas, las curvas solución recuerdan al modelo predador-presa de Volterra-Lotka.

(%t24)



(%o24)

Created with wxMaxima

II.4. A_{3x3} con un valor propio real y el resto valores propios complejos

Sea el sistema el que se muestra a continuación:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 0 & -2 & -2 \\ 4 & -2 & 2 \\ 4 & -2 & 2 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 101 \\ 201 \\ 100 \end{pmatrix}$$

$$\begin{cases} y_1 = -2y_2 - 2y_3 \\ y_2 = 4y_1 - 2y_2 + 2y_3 \\ y_3 = 4y_1 - 2y_2 + 2y_3 \end{cases}$$

En este caso resolveremos el sistema en dos etapas, en una trabajaremos con el valor propio real, en la otra con el valor propio compleja. Cargamos la librería SEDLPOH.mac, diagonalizamos la matriz de coeficientes y obtenemos los valores propios y vectores propios.

El procedimiento general que aplicaremos consiste en obtener primero las soluciones parciales **z1** y **z2** a partir de la raíz compleja, y en segundo lugar **z3** a partir del valor propio real. Por consiguiente, la solución del sistema es posible ya que se hace uso combinado de las técnicas descritas anteriormente.

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$
```

```
(%i2) A:matrix([0,-2,-2],[4,-2,2],[4,-2,2]);
```

$$(\%o2) \begin{bmatrix} 0 & -2 & -2 \\ 4 & -2 & 2 \\ 4 & -2 & 2 \end{bmatrix}$$

```
(%i3) charpoly(A,%lambda);
```

$$(\%o3) \quad -((-\lambda-2)(2-\lambda)+4)\lambda+2(4(2-\lambda)-8)-2(-4(-\lambda-2)-8)$$

```
(%i4) factor(%);
```

$$(\%o4) \quad -\lambda(\lambda^2+16)$$

```
(%i5) eigenvalues(A);
```

```
(%o5) [[-4 %i, 4 %i, 0], [1, 1, 1]]
```

```
(%i6) eigenvectors(A);
```

```
(%o6) [[[-4 %i, 4 %i, 0], [1, 1, 1]], [[[1, %i, %i]], [[1, -%i, -%i]], [[1, 1, -1]]
]]
```

Con el fin de ejercitarnos utilizaremos en el ejemplo el vector propio obtenido “a mano”, y no el sugerido por wxMaxima, aunque muy bien podría utilizarse este último:

```
(%i7) Re: transpose([0,1,1]);
```

$$(\%o7) \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$$

(%i8) Im:transpose ([1,0,0]);

$$(\%o8)\quad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

(%i9) v:(Re+Im*%i);

$$(\%o9)\quad \begin{bmatrix} \%i \\ 1 \\ 1 \end{bmatrix}$$

(%i10) %alpha:0;

(%o10) 0

(%i11) %beta:4;

(%o11) 4

En primer lugar trabajaremos con el valor propio complejo. Cargamos el paquete **SEDLPOHC.mac** con rutinas para el caso con valores propios complejos, obteniendo con **spahi$** las soluciones parciales **z1** y **z2**, la parte correspondiente de la **solución general del sistema homogéneo (SGHI)** con la orden **sghi**:

(%i12) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOHRC.mac")$

(%i13) spahi$

(%i14) z1;

$$(\%o14)\quad \begin{bmatrix} -\sin(4\,t) \\ \cos(4\,t) \\ \cos(4\,t) \end{bmatrix}$$

(%i15) z2;

$$(\%o15) \begin{bmatrix} \cos(4\,t) \\ \sin(4\,t) \\ \sin(4\,t) \end{bmatrix}$$

(%i16) sghi;

$$(\%o16) \begin{bmatrix} c2\cos(4\,t) - c1\sin(4\,t) \\ c2\sin(4\,t) + c1\cos(4\,t) \\ c2\sin(4\,t) + c1\cos(4\,t) \end{bmatrix}$$

En segundo lugar, continuaremos resolviendo el sistema trabajando ahora con el valor propio real. Introducimos el vector propio **v3**, obtenemos la solución parcial z_3 con la orden **spah(lambda, v**), y la parte de la **solución general de la homogénea** correspondiente **(SGHR)** por medio de la siguiente operación: **sghr:z3*c3**:

(%i17) v3:transpose([1,1,-1]);

$$(\%o17) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

(%i18) z3:spah(0,v3);

$$(\%o18) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}$$

(%i19) sghr:z3*c3;

$$(\%o19) \begin{bmatrix} c3 \\ c3 \\ -c3 \end{bmatrix}$$

En tercer lugar, construimos la **matriz fundamental G** reuniendo las soluciones parciales que hemos obtenido, evaluando su validez:

(%i20) G:addcol(z1,z2,z3);

$$(\%o20)\quad \begin{bmatrix} -\sin(4\,t) & \cos(4\,t) & 1 \\ \cos(4\,t) & \sin(4\,t) & 1 \\ \cos(4\,t) & \sin(4\,t) & -1 \end{bmatrix}$$

(%i21) rank(G);

$$(\%o21)\quad 3$$

(%i22) determinant(G);

$$(\%o22)\quad 2\,\sin(4\,t)^2+2\,\cos(4\,t)^2$$

En cuarto lugar, obtendremos la **solución general del sistema homogéneo** (SGH) reuniendo en **sgh** la **sghi** y **sghr**, incluido el paso adicional que se muestra en **%i24**:

(%i23) sgh:sghi+sghr;

$$(\%o23)\quad \begin{bmatrix} -c1\,\sin(4\,t)+c2\,\cos(4\,t)+c3 \\ c2\,\sin(4\,t)+c1\,\cos(4\,t)+c3 \\ c2\,\sin(4\,t)+c1\,\cos(4\,t)-c3 \end{bmatrix}$$

(%i24) sgh: list_matrix_entries(%);

$$(\%o24)\quad [-c1\,\sin(4\,t)+c2\,\cos(4\,t)+c3,\ c2\,\sin(4\,t)+c1\,\cos(4\,t)+c3,\ c2\,\sin(4\,t)+c1\,\cos(4\,t)-c3]$$

En quinto, y último lugar, introducimos el vector de condiciones iniciales Y0, obtenemos el valor de las constantes **c1**, **c2** y **c3**, que sustituiremos en la orden **sphec3(c1, c2, c3)** con la que se obtiene la **solución particular del sistema homogéneo** (SPH) de tres ecuaciones:

(%i25) Y0:transpose([101,201,100]);

(%o25) $\begin{bmatrix} 101 \\ 201 \\ 100 \end{bmatrix}$

(%i26) sisconstec3(Y0),numer;

```
rat: replaced -101.0 by -101/1 = -101.0
rat: replaced 1.0 by 1/1 = 1.0
rat: replaced -201.0 by -201/1 = -201.0
rat: replaced 1.0 by 1/1 = 1.0
rat: replaced -100.0 by -100/1 = -100.0
rat: replaced 1.0 by 1/1 = 1.0
```

(%o26) $[c1=150.5, c2=50.5, c3=50.5]$

(%i27) sphec3(150.5,50.5,50.5);

(%o27) $[-150.5\sin(4\,t)+50.5\cos(4\,t)+50.5, 50.5\sin(4\,t)+150.5\cos(4\,t)+50.5, 50.5\sin(4\,t)+150.5\cos(4\,t)-50.5]$

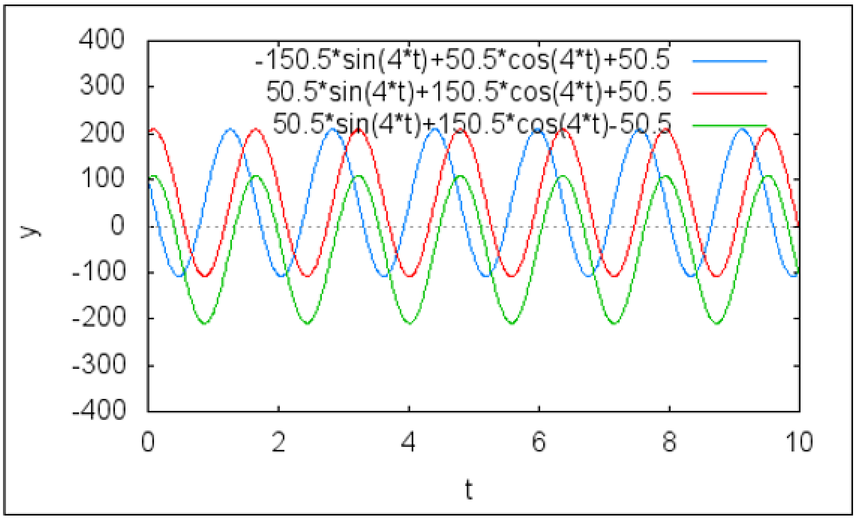
En este caso para que wxMaxima realice la gráfica de las curvas solución particular, efectuaremos los pasos siguientes:

(%i28) solpal:%;

(%o28) $[-150.5\sin(4\,t)+50.5\cos(4\,t)+50.5, 50.5\sin(4\,t)+150.5\cos(4\,t)+50.5, 50.5\sin(4\,t)+150.5\cos(4\,t)-50.5]$

(%i29) wxplot2d(solpal,[t,0,10],[y,-400,400]);

(%t29)



(%o29)

-->

Created with wxMaxima

II.5. A_{3x3} con dos valores propios reales y matriz de coeficientes no diagonalizable

Sea el sistema que se muestra a continuación:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 3 & 0 & -1 \\ 0 & 3 & 0 \\ 1 & -2 & 1 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 60 \\ 2 \\ 100 \end{pmatrix}$$

Se trata de una matriz A_{3x3} que **no es diagonalizable** por lo que deberemos “tantear y buscar el vector propio” que nos falta, y con el que se obtiene una de las soluciones parciales del sistema de ecuaciones homogéneo. En primer lugar, cargamos la librería SEDLPOH.mac, diagonalizamos la matriz de coeficientes y obtenemos los valores propios y vectores propios:

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$

(%i2) A:matrix([3,0,-1],[0,3,0],[1,-2,1]);
```

$$(\%o2)\quad \begin{bmatrix} 3 & 0 & -1 \\ 0 & 3 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

```
(%i3) charpoly(A,%lambda);
```

$$(\%o3)\quad -\lambda+(1-\lambda)(3-\lambda)^2+3$$

```
(%i4) factor(%);
```

$$(\%o4)\quad -(\lambda-3)(\lambda-2)^2$$

```
(%i5) eigenvalues(A);
```

$$(\%o5)\quad [[3,2],[1,2]]$$

```
(%i6) eigenvectors(A);
```

$$(\%o6)\quad [[[3,2],[1,2]],[[[1,\frac{1}{2},0]],[[1,0,1]]]]$$

En segundo lugar, con el valor propio con multiplicidad 1, esto es con $\lambda_1 = 3\ (m_1 = 1)$, obtenemos la primera solución parcial **z1** con la orden **spah**. Escribimos el vector propio **v1** sugerido por wxMaxima pero multiplicando por dos sus componentes. Esta operación es útil si resolviéramos el sistema "a mano", siendo anecdótica para la resolución con wxMaxima:

```
(%i7) v1:transpose([2,1,0]);
```

$$(\%o7)\quad \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$

(%i8) z1:spah(3,v1);

$$(\%o8) \begin{bmatrix} 2\ \%e^{3\,t} \\ \%e^{3\,t} \\ 0 \end{bmatrix}$$

Seguidamente y con el valor propio con multiplicidad 2, esto es con $\lambda_2 = 2\ (m_2 = 2)$, obtenemos aplicando un procedimiento similar al del primer valor propio la segunda solución parcial **z2**, escribiendo el vector propio **v2** sugerido por wxMaxima:

(%i9) v2:transpose([1,0,1]);

$$(\%o9) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

(%i10) z2:spah(2,v2);

$$(\%o10) \begin{bmatrix} \%e^{2\,t} \\ 0 \\ \%e^{2\,t} \end{bmatrix}$$

En tercer lugar, necesitamos un tercer vector propio, **buscándose por "tanteo"** empleando la expresión de la **Caja 5**. Esta tarea puede ser realizada con la orden **M(A, λ, n)** incluida en la librería SEDLPOH.mac

Caja 5

$$\left(A - \lambda\, I\right)^n \vec{V} = \vec{0}$$

En este caso, con la matriz de coeficientes A del ejemplo, $\lambda = 2$ y n=2, si realizáramos las operaciones "a mano" tendríamos que resolver el sistema compatible indeterminado que resulta de la siguiente matriz:

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & -2 & -1 \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & -2 & -1 \end{pmatrix} . \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & -2 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Con el fin de evitar operaciones "a mano" podemos hacer uso de wxMaxima tal y como se muestra en la **Caja 6**.

Caja 6

```
(%i1) A:matrix([3,0,-1],[0,3,0],[1,-2,1]);
```

(%o1) $\begin{bmatrix} 3 & 0 & -1 \\ 0 & 3 & 0 \\ 1 & -2 & 1 \end{bmatrix}$

```
i) MULTIPLICACION: utilizar operador asterisco: ·
```

```
(%i2) M:A-2·ident(3);
```

(%o2) $\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & -2 & -1 \end{bmatrix}$

```
ii) MULTIPLICACION: utilizar operador punto: .
```

```
(%i3) M.M;
```

(%o3) $\begin{bmatrix} 0 & 2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Observe que wxMaxima escribe los operadores ***** como 'punto arriba' y **.** como 'punto abajo'.

Basta observar la matriz para comprobar que la segunda componente del vector propio vale 0, mientras que la primera y tercera pueden tomar cualquier valor. Volviendo a la explicación del procedimiento, tendremos que llamando a la rutina y asignándole los parámetros de nuestro caso **M(A, 2, 2)**, obtendremos la estructura del vector propio que estamos buscando, **v3**:

```
(%i11) M(A,2,2);

solve: dependent equations eliminated: (3 2)
(%o11) [w1=%r2, w2=0, w3=%r1]
```

Puesto que hay dos posibles vectores propios que sean útiles para nuestro caso $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ y $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ elegimos por ejemplo el segundo de ellos.

Obtendremos la solución parcial **z3** con la orden **spah2(A, λ , v)** que la calcula a partir de la siguiente expresión (**Caja 7**):

Caja 7

$$z_i(t) = e^{\lambda t} . \left[\vec{V} + (A - \lambda I)\vec{V} t \right]$$

(%i12) v3:transpose([1,0,0]);

$$(\%o12) \quad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

(%i13) z3:spah2(A,2,v3);

$$(\%o13) \quad \begin{bmatrix} (t+1)\,\%e^{2\,t} \\ 0 \\ t\,\%e^{2\,t} \end{bmatrix}$$

En cuarto lugar, construimos la **matriz fundamental G** y comprobamos su validez. Puesto que el rango es 3, efectivamente los vectores son linealmente independientes continuando con el procedimiento que nos llevará a encontrar la solución del sistema homogéneo. Obtenemos **la solución general del sistema homogéneo** (SGH) de 3 ecuaciones con la orden **sghec3(G)**:

(%i14) G:addcol(z1,z2,z3);

$$(\%o14) \quad \begin{bmatrix} 2\,\%e^{3\,t} & \%e^{2\,t} & (t+1)\,\%e^{2\,t} \\ \%e^{3\,t} & 0 & 0 \\ 0 & \%e^{2\,t} & t\,\%e^{2\,t} \end{bmatrix}$$

(%i15) rank(G);

(%o15) 3

(%i16) determinant(G);

(%o16) $(t+1)\%e^{7t}-t\%e^{7t}$

(%i17) sghec3(G);

(%o17) $$\begin{bmatrix} 2\,c1\,\%e^{3t}+c3\,(t+1)\,\%e^{2t}+c2\,\%e^{2t} \\ c1\,\%e^{3t} \\ c3\,t\,\%e^{2t}+c2\,\%e^{2t} \end{bmatrix}$$

Finalmente, introducimos el vector de condiciones iniciales Y0, obtenemos los valores de las constantes **c1**, **c2** y **c3**, y concluimos ejecutando la orden **sphec3(c1, c2, c3)** con la que se obtiene la **solución particular del sistema homogéneo** (SPH):

(%i18) Y0:transpose([60,2,100]);

(%o18) $$\begin{bmatrix} 60 \\ 2 \\ 100 \end{bmatrix}$$

(%i19) sisconstec3(Y0);

(%o19) $[c1=2, c2=100, c3=-44]$

(%i20) sphec3(2,100,-44);

(%o20) $$\begin{bmatrix} 4\,\%e^{3t}-44\,(t+1)\,\%e^{2t}+100\,\%e^{2t} \\ 2\,\%e^{3t} \\ 100\,\%e^{2t}-44\,t\,\%e^{2t} \end{bmatrix}$$

La representación gráfica de las curvas solución particular es efectuada aplicando el procedimiento descrito anteriormente:

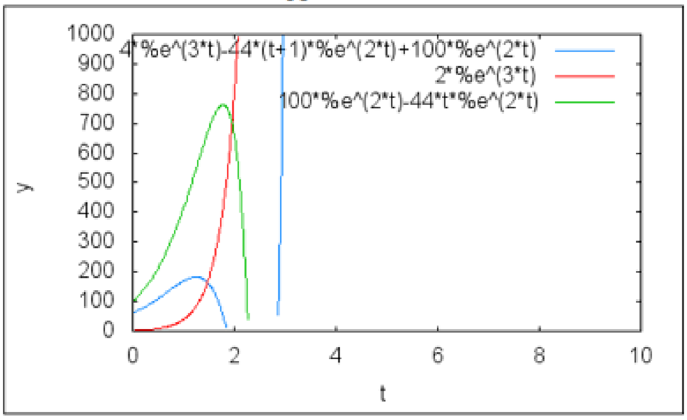
```
(%i21) solpal:list_matrix_entries(%);
```

(%o21) $[4\,\%e^{3\,t}-44\,(t+1)\,\%e^{2\,t}+100\,\%e^{2\,t}, 2\,\%e^{3\,t}, 100\,\%e^{2\,t}-44\,t\,\%e^{2\,t}]$

```
(%i22) wxplot2d(solpal,[t,0,10],[y,0,1000]);
```

```
plot2d: some values were clipped.
plot2d: some values were clipped.
plot2d: some values were clipped.
```

(%t22)



(%o22)

Created with wxMaxima.

II.6. A_{3x3} con un valor propio real y matriz de coeficientes no diagonalizable

Resolveremos el sistema que se muestra a continuación:

$$Y' = A.Y$$

$$A = \begin{pmatrix} 4 & 5 & -2 \\ -2 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix} \quad , \quad Y(0) = \begin{pmatrix} 60 \\ 278 \\ 10 \end{pmatrix}$$

Se trata de un caso relacionado con **II.5** ya que la matriz A_{3x3} **no es diagonalizable**, excepto que ahora deberemos "tantear y buscar dos vectores propios" en lugar de uno como en el caso anterior. En primer lugar, cargamos la librería SEDLPOH.mac, diagonalizamos la matriz de coeficientes y obtenemos los valores propios y vectores propios:

```
(%i1) load("C:/Users/Rafa Lahoz-Beltra/Desktop/Working Area/PracticasMates/SEDLPOH.mac")$

(%i2) A:matrix([4,5,-2],[-2,-2,1],[-1,-1,1]);
```

(%o2) $\begin{bmatrix} 4 & 5 & -2 \\ -2 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$

```
(%i3) charpoly(A,%lambda);
```

(%o3) $2\,\lambda+((-\lambda-2)(1-\lambda)+1)(4-\lambda)-5(1-2(1-\lambda))$

```
(%i4) factor(%);
```

(%o4) $-(\lambda-1)^3$

```
(%i5) eigenvalues(A);
```

(%o5) [[1],[3]]

En segundo lugar, introducimos el único vector propio proporcionado por wxMaxima al obtenerse un único valor propio $\lambda=1$ con multiplicidad 3, construyendo la primera solución parcial **z1**:

```
(%i7) v1:transpose([1,-1,-1]);
```

(%o7) $\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}$

```
(%i8) z1:spah(1,v1);
```

(%o8) $\begin{bmatrix} \%e^{t} \\ -\%e^{t} \\ -\%e^{t} \end{bmatrix}$

En tercer lugar, y a partir del mismo planteamiento teórico expuesto en el ejemplo anterior (1.5) en este caso con la matriz de coeficientes A del ejemplo, $\lambda = 1$ y n=2 tanteamos con **M(A, 1, 2)** buscando alguno de los vectores propios de los dos que nos faltan. En este caso, y si realizamos las operaciones "a mano":

$$\begin{pmatrix} 3 & 5 & -2 \\ -2 & -3 & 1 \\ -1 & -1 & 0 \end{pmatrix}^2 = \begin{pmatrix} 1 & 2 & -1 \\ -1 & -2 & 1 \\ -1 & -2 & 1 \end{pmatrix}$$

obtenemos de $(A-I)^2 \vec{V} = \vec{0}$ un sistema de ecuaciones tal que v1 +2v2 –v3 =0 deduciendo un vector propio con estructura $\begin{pmatrix} \alpha \\ \beta \\ \alpha + 2\beta \end{pmatrix}$ ya que v3=v1+2v2. Asignando valores 1,0 y 0,1 a α, β conseguimos dos vectores propios, $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$ y $\begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$ respectivamente. Observe que wxMaxima nos devuelve un vector propio con estructura equivalente $\begin{pmatrix} \alpha - 2\beta \\ \beta \\ \alpha \end{pmatrix}$, tal que para distintos valores de los parámetros tenemos los vectores $\begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}$ y $\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$. Utilizaremos los dos vectores propios **v2** y **v3** que sugiere wxMaxima, obteniendo las soluciones parciales **z2** y **z3** para cada vector con la orden **spah2(A, λ, v)**. Recuérdese que ésta orden calcula cada solución parcial utilizando la siguiente expresión:

$$z_i(t) = e^{\lambda t}.\left[\vec{V} + (A - \lambda I)\vec{V}t\right]$$

```
(%i9) M(A,1,2);

solve: dependent equations eliminated: (2 3)
(%o9) [w1=%r1-2 %r2, w2=%r2, w3=%r1]

(%i10) v2:transpose([1,0,1]);
```

$$(\%o10) \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

(%i11) z2:spah2(A,1,v2);

$$(\%o11) \begin{bmatrix} (t+1)\,\%e^{t} \\ -t\,\%e^{t} \\ (1-t)\,\%e^{t} \end{bmatrix}$$

(%i12) v3:transpose([-2,1,0]);

$$(\%o12) \begin{bmatrix} -2 \\ 1 \\ 0 \end{bmatrix}$$

(%i13) z3:spah2(A,1,v3);

$$(\%o13) \begin{bmatrix} (-t-2)\,\%e^{t} \\ (t+1)\,\%e^{t} \\ t\,\%e^{t} \end{bmatrix}$$

En cuarto lugar, construimos la **matriz fundamental G** y comprobamos su validez. Puesto que el **rango es 2**, la matriz G no es válida. Más aún, si simplificamos la expresión del determinante de la matriz G con la orden **ratsimp(determinant(G))** de wxMaxima comprobamos que su valor es 0:

(%i14) G:addcol(z1,z2,z3);

$$(\%o14) \begin{bmatrix} \%e^{t} & (t+1)\,\%e^{t} & (-t-2)\,\%e^{t} \\ -\%e^{t} & -t\,\%e^{t} & (t+1)\,\%e^{t} \\ -\%e^{t} & (1-t)\,\%e^{t} & t\,\%e^{t} \end{bmatrix}$$

```
(%i15) rank(G);

(%o15) 2

(%i16) ratsimp(determinant(G));

(%o16) 0
```

Por consiguiente, alguno de los vectores propios que hemos tanteado debe ser eliminado, continuando con la búsqueda de un nuevo vector propio. Por ejemplo, **eliminaremos el tercer vector**, **v3**, y por tanto **z3** y la tercera columna de la matriz fundamental G. El tanteo de un nuevo vector propio requiere ahora que $\left(A-I\right)^3\vec{V}=\vec{0}$. Obsérvese que la estructura del vector propio que devuelve la orden **M(A, 1, 3)** es tal que "cualquier" vector elegido es válido siempre y cuando sea distinto de los dos que hemos utilizado para construir la primera y segunda columnas de la matriz fundamental G. Esto se confirma con la **matriz nula** que resulta de efectuar la operación **(A-1*ident(3))^^3** que realizamos con el fin de comprobar el resultado en wxMaxima:

```
(%i17) M(A,1,3);

solve: dependent equations eliminated: (1 3 2)
(%o17) [w1=%r5, w2=%r4, w3=%r3]

(%i18) (A-1*ident(3))^^3;
```

$$(\%o18)\quad \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

En consecuencia ¿qué vector propio elegimos como **v3**? Por ejemplo, el vector $\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, obteniendo con el vector elegido la solución parcial **z3**. Puesto que estamos tanteando por segunda vez la búsqueda de un vector propio, utilizaremos la orden **spah3(A, λ, v)** que calcula la solución parcial a partir de la siguiente expresión:

$$z_i(t)=e^{\lambda t}.\left[\vec{V}+(A-\lambda I)\vec{V}t+(A-\lambda I)^2\vec{V}\frac{t^2}{2!}\right]$$

Construimos de nuevo la **matriz fundamental G** con las soluciones parciales que obtuvimos anteriormente, **z1** y **z2**, y la nueva solución **z3**, preguntándonos una vez más sobre su validez. Obviamente G debe ser válida en congruencia con la matriz nula del paso **%o18** al indicarnos éste último resultado que las componentes del vector propio son independientes unas de otras, pudiendo tomar cualquier valor con tal de que el vector no sea $\vec{0}$. Finalmente, comprobamos la validez de la matriz G ya que el rango es 3 y la expresión simplificada del determinante *–exp(3t)*, es decir distinta de cero (**%o23**):

```
(%i19) v3:transpose([1,0,0]);
```

$$(\%o19)\quad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

```
(%i20) z3:spah3(A,1,v3);
```

$$(\%o20)\quad \begin{bmatrix} \left(\frac{t^2}{2}+3\ t+1\right)\%e^{t} \\ \left(-\frac{t^2}{2}-2\ t\right)\%e^{t} \\ \left(-\frac{t^2}{2}-t\right)\%e^{t} \end{bmatrix}$$

```
(%i21) G:addcol(z1,z2,z3);
```

$$(\%o21)\quad \begin{bmatrix} \%e^{t} & (t+1)\ \%e^{t} & \left(\frac{t^2}{2}+3\ t+1\right)\%e^{t} \\ -\%e^{t} & -t\ \%e^{t} & \left(-\frac{t^2}{2}-2\ t\right)\%e^{t} \\ -\%e^{t} & (1-t)\ \%e^{t} & \left(-\frac{t^2}{2}-t\right)\%e^{t} \end{bmatrix}$$

```
(%i22) rank(G);
```

$$(\%o22)\quad 3$$

```
(%i23) ratsimp(determinant(G));
```

$$(\%o23)\quad -\%e^{3\ t}$$

En quinto lugar, la **solución general del sistema homogéneo** (SGH) es obtenida con la orden **sghec3(G)**:

```
(%i24) sghec3(G);
```

$$(\%o24) \begin{bmatrix} c3\left(\frac{t^2}{2}+3\,t+1\right)\%e^{t}+c2\,(t+1)\,\%e^{t}+c1\,\%e^{t} \\ c3\left(-\frac{t^2}{2}-2\,t\right)\%e^{t}-c2\,t\,\%e^{t}-c1\,\%e^{t} \\ c3\left(-\frac{t^2}{2}-t\right)\%e^{t}+c2\,(1-t)\,\%e^{t}-c1\,\%e^{t} \end{bmatrix}$$

Finalmente, en sexto y último lugar, encontraremos la **solución particular del sistema homogéneo** (SPH) como en ejemplos anteriores. Introducimos el **vector de condiciones iniciales Y0**, resolvemos el sistema de constantes con la orden **sisconstec3(Y0)** siendo la solución el parámetro de la orden **sphec3(c1, c2, c3)**:

```
(%i25) Y0:transpose([60,278,10]);
```

$$(\%o25) \begin{bmatrix} 60 \\ 278 \\ 10 \end{bmatrix}$$

```
(%i26) sisconstec3(Y0);
```

$$(\%o26) \; [\,c1=-278\,,\; c2=-268\,,\; c3=606\,]$$

```
(%i27) sphec3(-278,-268,606);
```

$$(\%o27) \begin{bmatrix} 606\left(\frac{t^2}{2}+3\,t+1\right)\%e^{t}-268\,(t+1)\,\%e^{t}-278\,\%e^{t} \\ 606\left(-\frac{t^2}{2}-2\,t\right)\%e^{t}+268\,t\,\%e^{t}+278\,\%e^{t} \\ 606\left(-\frac{t^2}{2}-t\right)\%e^{t}-268\,(1-t)\,\%e^{t}+278\,\%e^{t} \end{bmatrix}$$

Concluiremos el protocolo de resolución del sistema representando gráficamente las curvas solución particular:

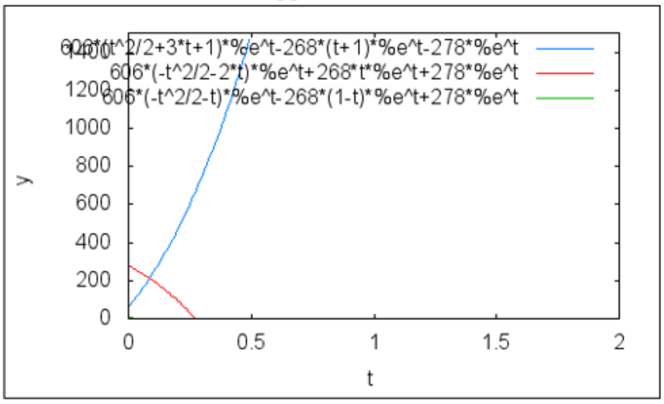
(%i28) solpal:list_matrix_entries(%);

(%o28) $[606\left(\frac{t^2}{2}+3\,t+1\right)\%e^t-268\,(t+1)\,\%e^t-278\,\%e^t,\ 606\left(-\frac{t^2}{2}-2\,t\right)\%e^t+268\,t\,\%e^t+278\,\%e^t,\ 606\left(-\frac{t^2}{2}-t\right)\%e^t-268\,(1-t)\,\%e^t+278\,\%e^t]$

(%i29) wxplot2d(solpal,[t,0,2],[y,0,1500]);

```
plot2d: some values were clipped.
plot2d: some values were clipped.
plot2d: some values were clipped.
```

(%t29)



(%o29)

-->

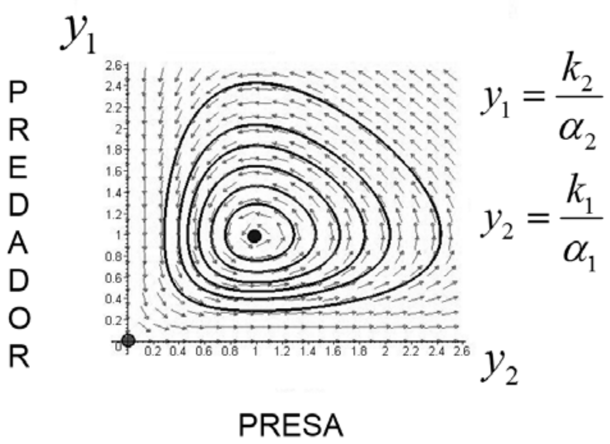
Created with wxMaxima

III. Estudio con wxMaxima de sistemas de ecuaciones diferenciales no lineales: el modelo predador-presa de Volterra-Lotka

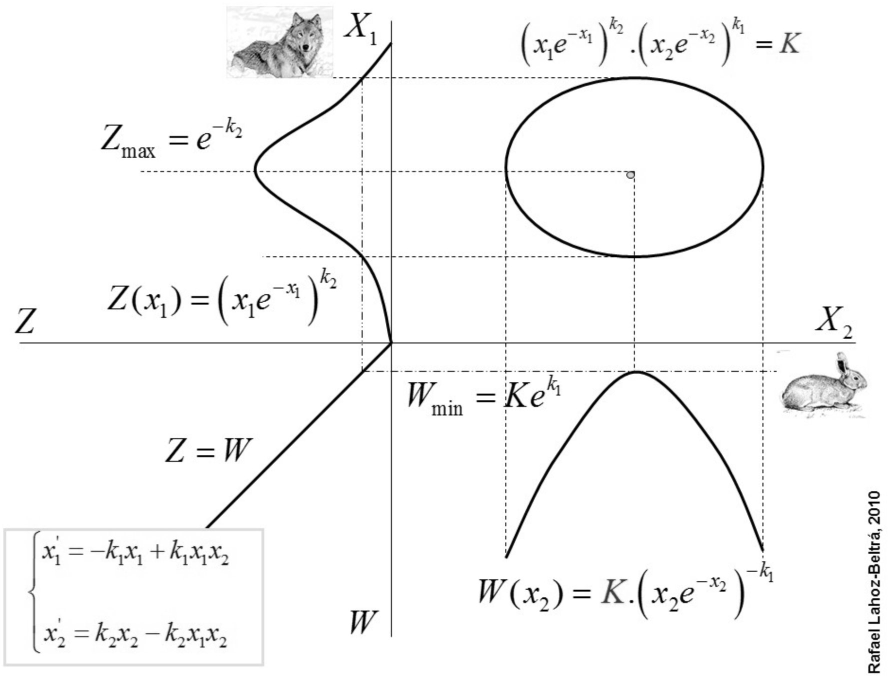
Estudiaremos el sistema de ecuaciones diferenciales propuesto por V. Volterra y A. Lotka para modelar la coexistencia de dos poblaciones, una de predadores (y_1) y otra de presas (y_2):

$$\begin{cases} y_1' = -k_1 y_1 + \alpha_1 y_1 y_2 \\ y_2' = k_2 y_2 - \alpha_2 y_1 y_2 \end{cases}$$

siendo k_1 la tasa de mortalidad de los predadores y k_2 la tasa de crecimiento de las presas. Los coeficientes α_1 y α_2 modelan la interacción entre ambas poblaciones. En el modelo los **puntos de equilibrio** son:



Una de las vías para resolver el sistema es hacer un cambio de variable, tal que podremos "trabajar" en situaciones prácticas ya sea con el ordenador o "a mano" a partir de la siguiente figura:



Estudiaremos el modelo de Volterra-Lotka con wxMaxima. Supóngase dos poblaciones en interacción, una de predadores y otra de presas, con los siguientes parámetros del modelo y condiciones iniciales, obteniéndose en (%o5) y (%o6) las coordenadas del **punto de equilibrio**:

```
Parámetros del modelo

(%i1) k1: 0.3;
(%o1) 0.3

(%i2) alfa1: 0.001;
(%o2) 0.001

(%i3) k2: 0.8;
(%o3) 0.8

(%i4) alfa2: 0.02;
(%o4) 0.02

(%i5) PE_predadores: k2/alfa2;
(%o5) 40.0

(%i6) PE_presas: k1/alfa1;
(%o6) 300.0

Condiciones iniciales

(%i7) y01: 35;
(%o7) 35

(%i8) y02: 420;
(%o8) 420
```

Aplicando el **método de Runga-Kutta de 4º orden**:

```
Sistema de Ecuaciones Diferenciales

(%i9) predadores: -k1·y1 + alfa1·y1·y2;
(%o9) 0.001 y1 y2-0.3 y1

(%i10) presas: k2·y2-alfa2·y1·y2;
(%o10) 0.8 y2-0.02 y1 y2

(%i11) fll(x) := [first(x),last(x),length(x)]$ declare(fll,evfun)$

(%i13) puntos: rk([predadores, presas],[y1, y2], [y01, y02], [t, 0, 100, 0.1]);
```

obtendremos:

```
(%o13) [[0,35,420],[0.1,35.42978213227254,424.0401627118749],[0.2,35.87873753521
36.83131914499392,434.0354691120201],[0.5,37.33318764036326,436.576864314421],[0.
],[0.8,38.92755155894967,441.5290933259806],[0.9,39.4839554493789,442.23180127576
442.140780536643],[1.2,41.20432591641254,441.3333910074384],[1.3,41.7882381248302
42.95854632317789,435.8568765411642],[1.6,43.54017933310593,433.0332290502412],[1
],[1.9,45.24102755081493,421.756555496666],[2.0,45.78479785258209,417.13087141275
406.7347118524434],[2.3,47.31150052542176,401.02492223509],[2.4,47.77764589263058
48.63287560710719,382.2494452449005],[2.7,49.01820013233275,375.5596753546969],[2
],[3.0,49.98650609164511,354.694780592205],[3.1,50.24281816751275,347.58964644229
333.3483301580279],[3.4,50.80228288548816,326.2711223914277],[3.5,50.918033266387
51.04400530731417,305.5250648914612],[3.8,51.05510139264429,298.8455941955652],[3
```

Una vez obtenidos los puntos, ejecutaremos las siguientes órdenes en wxMaxima:

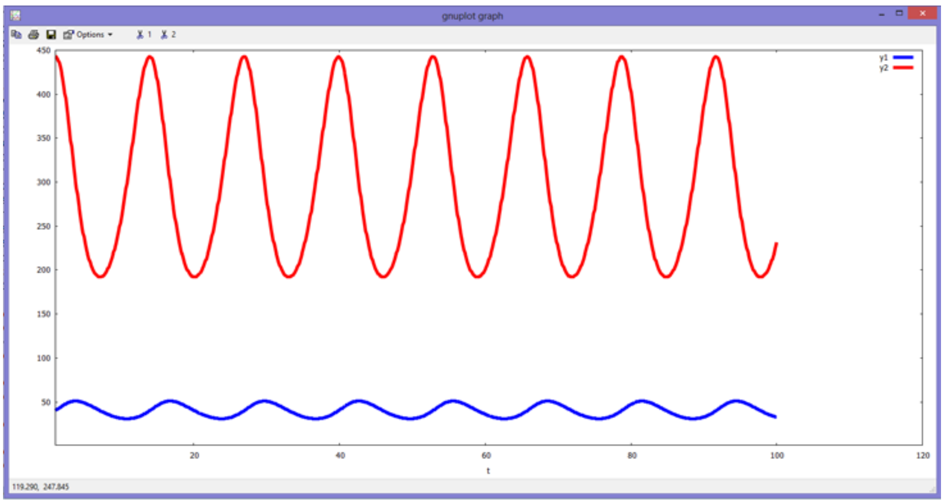
```
(%i14) %, fll;
(%o14) [[0,35,420],[100.0,32.42172383346249,231.0036158398463],1001]

(%i15) predadoresL: makelist([puntos[i][1], puntos[i][2]], i, 1, length(puntos))$

(%i16) %,fll;
(%o16) [[0,35],[100.0,32.42172383346249],1001]

(%i17) presasL: makelist([puntos[i][1], puntos[i][3]], i, 1, length(puntos))$

(%i18) %, fll;
(%o18) [[0,420],[100.0,231.0036158398463],1001]
```

Ahora, ya estaremos en condiciones de obtener con wxMaxima las curvas clásicas del modelo de Volterra-Lotka. En primer lugar, y con el fin de obtener la **curva de predadores y presas** escribiremos la orden que se muestra a continuación (NOTA: escribir en una sola línea):

```
Curvas de predadores y presas
```

(%i41) plot2d([[discrete, predadoresL],[discrete, presasL]], [x, 1, 120],[style, [lines, 5]], [y, 1, 450], [ylabel, " "], [xlabel, "t"], [legend, "y1", "y2"])$

obteniendo:

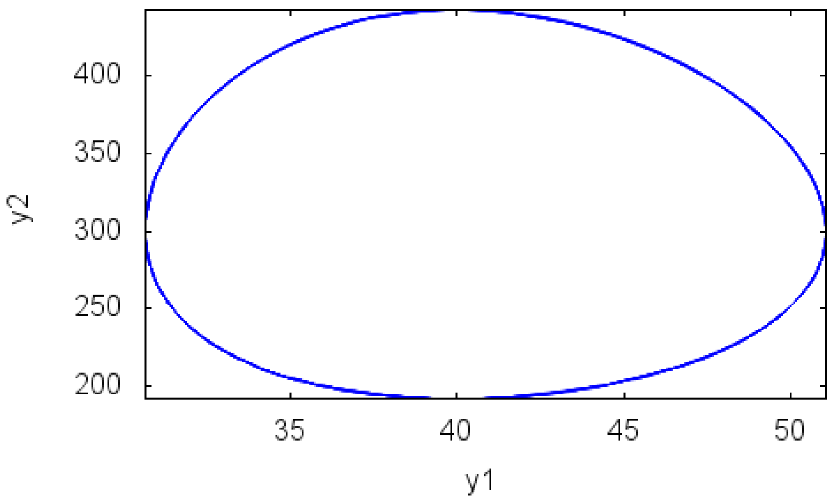


R. Lahoz-Beltra. (2014). *Métodos en Biomatemática II: Ecuaciones Diferenciales con wxMaxima*. eprints.ucm.es/26851/

En la figura ¿qué curva corresponde a los predadores y a las presas?

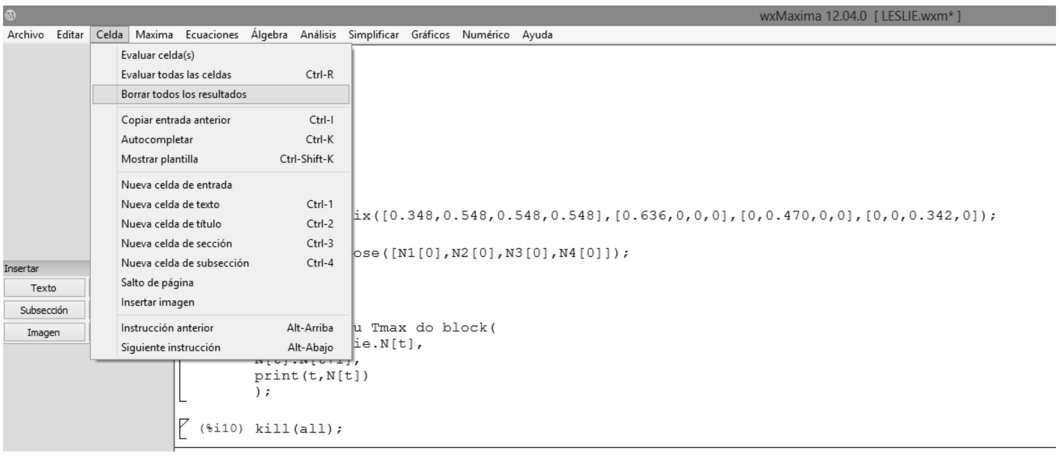
En segundo lugar, representaremos el **ciclo solución**:

```
Ciclo solución

 -->  puntosL: map(lambda([x],rest(x)),puntos)$

 -->  wxdraw2d(point_type=none, points_joined=true, xlabel="y1", ylabel="y2", points(puntosL));
```



IV. APENDICE

Es importante recordar que la computación en wxMaxima con vectores y matrices requiere **Reiniciar wxMaxima** y **Limpiar memoria** con cierta frecuencia, antes de ejecutar las órdenes. Estas opciones están disponibles en **wxMaxima** en la barra de tareas. También deberían **borrarse todos los resultados** previos, opción disponible en **Celda**:



Pantalla de wxMaxima mostrando la tarea 'Borrar todos los resultados'

Los ejemplos han sido resueltos utilizando las **librerías de rutinas SEDLPOH.mac** (**Caja 8**) y **SEDLPOHRC.mac** (**Caja 9**) simplificándose notablemente el trabajo. La primera librería es una colección de rutinas útiles cuando las raíces del polinomio característico son □ mientras que la segunda librería simplifica los cálculos en el caso de que el polinomio característico tenga raíces □ .

Caja 8

```
/*
Práctica. Laboratorio de cálculo. Librería SEDLPOH.mac
*/

/*
 Rafael Lahoz-Beltra. Departamento de Matemática Aplicada (Biomatemática). Facultad de Biología. UCM.
*/

/*
Polinomio con raices reales A3x3: (1) Distintas: lambda1 (m1=1), lambda2 (m2=1), lambda3 (m3=1) (2) lambda1 (m1=1) y lambda2 (m2=2) (3) lambda1 (m1=3)
*/
M(A,%lambda,n):=block(w:matrix([w1],[w2],[w3]),linsolve(list_matrix_entries((A-%lambda*ident(3))^^n.w),[w1,w2,w3]))$
spah(%lambda,v):=%e^(%lambda*t)*v$
spah2(A,%lambda,v):= block ( exp(%lambda*t)*(v + (A-%lambda*ident(3)).v*t))$
spah3(A,%lambda,v):=block ( exp(%lambda*t)*(( v + (A-%lambda*ident(3)).v*t) + (((A-%lambda*ident(3))^^2).v*(t^2)/2)))$
sghec3(G):= G.transpose([c1,c2,c3])$
sphec3(c01,c02,c03):=subst([c1=c01,c2=c02,c3=c03],sghec3(G))$
sisconstec3(Y0):=block(se:subst(0, t, sghec3(G))-Y0, sistecs:list_matrix_entries(se), linsolve(sistecs,[c1,c2,c3]))$

/*
Polinomio característico con raices reales A2x2:
*/
sghec2(G):=G.transpose([c1,c2])$
sphec2(c01,c02):=subst([c1=c01,c2=c02],sghec2(G))$
sisconstec2(Y0):=block(se:subst(0, t, sghec2(G))-Y0, sistecs:list_matrix_entries(se), linsolve(sistecs,[c1,c2]))$
```

El contenido de este manual es principalmente docente aunque las técnicas y métodos que se describan sean también de utilidad en la realización de proyectos. Los ejemplos han sido elegidos de diferentes fuentes con una finalidad didáctica siendo resueltos por el autor de este manual con wxMaxima. Su autor no se responsabiliza de posibles perjuicios derivados de cualquier otro uso de este manual fuera del ámbito para el que ha sido elaborado.

Caja 9

```
/*
Práctica. Laboratorio de cálculo. Librería SEDLPOHRC.mac
*/
```

```
/*
Rafael Lahoz-Beltra. Departamento de Matemática Aplicada (Biomatemática). Facultad de Biología. UCM.
*/

/*
Polinomio característico con raices complejas A2x2:
*/
spah_i(%alpha,%beta,Re,Im):=block(z1:exp(%alpha*t)*(Re*cos(%beta*t)-Im*sin(%beta*t)),
z2:exp(%alpha*t)*(Im*cos(%beta*t)+Re*sin(%beta*t)))$
spahi:spah_i(%alpha,%beta,Re,Im)$
sghi:c1*z1+c2*z2$
sisconsti(Y0):=block(se:subst(0, t, sghi)-Y0, sistecs:list_matrix_entries(se), linsolve(sistecs,[c1,c2]))$
sphi(c01,c02):=subst([c1=c01,c2=c02],sghi)$

/*
Resolución de sistema de tres ecuaciones de constantes para el caso de un polinomio característico con una raiz real y la otra raiz compleja.
*/
sisconstec3(Y0):=block(se:subst(0, t, sgh-Y0), sistecs:list_matrix_entries(se), linsolve(sistecs,[c1,c2,c3]))$
sphec3(c01,c02,c03):=subst([c1=c01,c2=c02,c3=c03],sgh)$
```

Bibliografía

[1] A. Vodopivec et al. 2011. wxMaxima. http://wxmaxima.sourceforge.net/

[2] R. Lahoz-Beltra. 2010. Las Matemáticas de la Vida. Modelos numéricos para la biología y la ecología. Colección “El Mundo es Matemático” RBA.

[3] R. Lahoz-Beltra. 2004. Bioinformática. Simulación, vida artificial e inteligencia artificial. Ediciones Díaz de Santos.

[4] M. C. Martínez Calvo, A. Pérez de Vargas. 1993. Métodos Matemáticos en Biología. Editorial Centro de Estudios Ramón Areces.

[5] M. C. Martínez Calvo, A. Pérez de Vargas. 1995. Problemas de Biomatemática. Editorial Centro de Estudios Ramón Areces.

[6] R. Lahoz-Beltra, J. Ortega, C. Fernández-Montraveta. 1994. Métodos Estadísticos en Biología del Comportamiento. Editorial Complutense.